Digital Audio Signature for 3D Printing Integrity

Sofia Belikovetsky, Yosef A. Solewicz, Mark Yampolskiy, Jinghui Toh, and Yuval Elovici

***Abstract*— Additive manufacturing (AM, or 3D printing) is a novel manufacturing technology that has been adopted in industrial and consumer settings. However, the reliance of this technology on computerization has raised various security concerns. In this paper, we address issues associated with sabotage via tampering during the 3D printing process by presenting an approach that can verify the integrity of a 3D printed object. Our approach operates on acoustic side-channel emanations generated by the 3D printer's stepper motors, which results in a non-intrusive and real-time validation process that is difficult to compromise. The proposed approach constitutes two algorithms. The first algorithm is used to generate a master audio fingerprint for the verifiable unaltered printing process. The second algorithm is applied when the same 3D object is printed again, and this algorithm validates the monitored 3D printing process by assessing the similarity of its audio signature with the master audio fingerprint. To evaluate the quality of the proposed thresholds, we identify the detectability thresholds for the following minimal tampering primitives: insertion, deletion, replacement, and modification of a single tool path command. By detecting the deviation at the time of occurrence, we can stop the printing process for compromised objects, thus saving time and preventing material waste. We discuss various factors that impact the method, such as background noise, audio device changes, and different audio recorder positions.**

***Index Terms*— Additive manufacturing, cyber security, side channels.**

I. Introduction

Additive manufacturing (AM), which is often referred to as 3D printing, is a manufacturing technology that creates parts and prototypes by incrementally fusing layers of material together. This manufacturing technology can create objects from polymers, metals, alloys, and composites.

AM has numerous technological, environmental, and socioeconomic advantages, such as the ability to manufacture objects with complex internal structures, shorter design-to-production times, just-in-time and on-demand production, and reduced raw material waste. These advantages enable a broad range of applications from generating models and prototypes to fabricating functional parts in safety-critical systems. A recent example of the latter is the FAA (Federal Aviation Administration)-approved 3D printed fuel nozzle for GE's state-of-the-art LEAP jet engine [1].

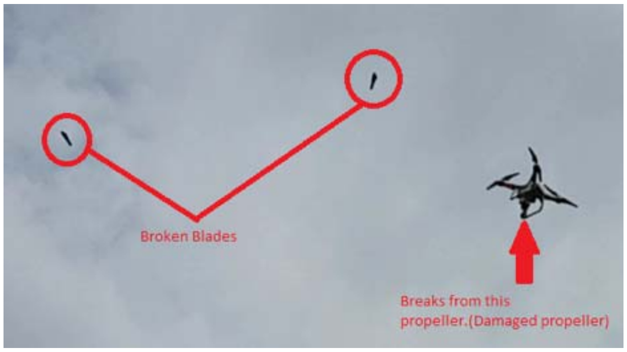


Fig. 1. Sabotaged quadcopter's propeller breaks during flight (*dr0wned* study [9]).

According to the Wohlers Report in 2016 [2], the AM industry accounted for $6.063 billion of revenue, with 33.8% of all AM-manufactured objects used as functional parts. A study conducted by Ernst and Young [3] showed that the adoption of this technology is rapidly increasing worldwide. In the U.S. alone, 16% of surveyed companies have had some experience with AM, and another 16% are considering adopting this technology in the future.

Due to the growing importance of AM and its reliance on computerization, many researchers have raised security concerns. Thus far, two major threat categories have been identified: (1) sabotage [4]–[10] and (2) violation of intellectual property (IP) rights [5], [11]–[13]. Sabotage attacks aim to inflict physical damage, such as by compromising part quality or damaging AM equipment. IP violation attacks aim to illegally replicate 3D objects or the manufacturing process itself. Additionally, several articles have discussed using 3D printers to manufacture illegal items, such as firearms or components of explosive devices [14]–[16].

This paper focuses exclusively on sabotage attacks and proposes a method of detecting such attacks. The importance of combating these attacks is illustrated by the recent *dr0wned* study [9], in which researchers presented a full chain of attack with AM and introduced a novel cyber-physical attack that caused material fatigue of a functional part. The authors sabotaged the 3D printed propeller of a quadcopter UAV, causing the propeller to break and the quadcopter to fall from the sky after a short period of flight (see Figure 1). Although

Manuscript received October 24, 2017; revised March 4, 2018 and May 18, 2018; accepted June 14, 2018. Date of publication July 12, 2018; date of current version January 23, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Anthony Ho. *(Corresponding author: Sofia Belikovetsky.)*

S. Belikovetsky and Y. A. Solewicz are with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beersheba 8410501, Israel, and also with the Cyber Security Research Center, Ben-Gurion University of the Negev, Beersheba 8410501, Israel (e-mail: sofiabel@post.bgu.ac.il).

M. Yampolskiy is with the Department of Computer Science, University of South Alabama, Mobile, AL 36688 USA.

J. Toh is with the iTrust Center of Research in Cyber Security, Singapore University of Technology and Design, Singapore 487372.

Y. Elovici is with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beersheba 8410501, Israel, and also with the Cyber Security Research Center, Ben-Gurion University of the Negev, Beersheba 8410501, Israel.

Digital Object Identifier 10.1109/TIFS.2018.2851584

1556-6013 © 2018 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

this sabotage only led to the loss of a $1000 drone, similar attacks on functional parts for safety-critical systems may cause tremendous monetary losses, disruptions, and loss of human life.

In this paper, we propose a method capable of detecting such cyber-physical attacks. Similar to the works of Chhetri *et al.* [17] and Bayens *et al.* [18], we exploit the fact that acoustic emanations from the fused deposition modeling (FDM) 3D printing process can be directly tied to the activities of all motors. However, our proposal has distinct differences. First, we do not use any specialized equipment. Instead, we perform all recording using a smartphone, thereby removing a significant hurdle and enabling the easy adoption of the proposed strategy by both industrial and home users. Furthermore, the verification algorithm can be implemented as a cloud-based app, which strengthens the security aspects of this solution. Second, we propose an entirely new approach for generating and verifying the audio signature of a 3D printing process. Our approach can detect deviations of individual G-Code commands representing manufacturing actions at a granular scale, which enables a significantly higher detectability rate than the 77.45% accuracy reported by Chhetri *et al.* [17] and the detection of minor deviations from the original design in addition to changes in the infill pattern, which was researched by Bayens *et al.* [18].

The remainder of the paper is structured as follows. After discussing previous work in this field in section II, we present the considered threat model in section III. Next, we introduce the proposed solution in section IV. An evaluation of the detection capabilities of the solution and its limitations is provided in section V, and a summary of the detection thresholds and the applicability to other known attacks is presented in section VI. The paper is concluded in section VII.

II. Related Work

By the end of 2017, approximately 70 (mostly peer-reviewed) publications addressed all three threat categories of AM security: theft of technical data, sabotage attacks, and manufacturing of illegal objects [19]. For this paper, the demonstrations of various sabotage attacks and proposals of attack detection techniques are relevant.

To the best of our knowledge, the first proof-of-concept showing the compromise of a desktop 3D printer was presented at XCon2013[1] by Hang [20]. The keynote speaker argued that the size (and thus integrability) of a printed part can be modified and that the temperature of the filament extruder can be manipulated, among other issues.

Several studies have analyzed 3D printers and 3D printing processes for vulnerabilities. Turner *et al.* [21] found that networking and communication systems lack integrity checks when receiving design files. Moore *et al.* [22] identified numerous vulnerabilities in software, firmware, and communication protocols commonly used in desktop 3D printers that could potentially be exploited. Do *et al.* [23] showed that communication protocols employed by desktop 3D printers can be exploited, thus enabling the retrieval of current and previously printed 3D models, cessation of an active printing job, or submission of a new job. Belikovetsky *et al.* [9] used a phishing attack to install a backdoor that enabled targeted manipulations of design files by a remote adversary. Sturm *et al.* [4] used malware pre-installed on a computer to automate the manipulation of STL files. Moore *et al.* 2016 [10] used malicious firmware to modify or substitute a printed 3D model.

A growing body of research discusses how a manufactured part's quality can be compromised. The bulk of the studies focuses on FDM, commonly used in desktop 3D printers. Sturm *et al.* [4] demonstrated that a part's tensile strength can be degraded by introducing defects such as voids (internal cavities). Zeltmann *et al.* [7] showed that similar results can be achieved by printing part of the structure with contaminated material. Belikovetsky *et al.* [9] proposed the degradation of a part's fatigue life and argued that the defect's size, geometry, and location are factors in the degradation. Yampolskiy *et al.* [6] argued that the anisotropy of 3D printed parts can be misused to degrade a part's quality if an object is printed in the wrong orientation. Zeltmann *et al.* [7] experimentally showed the impact of this attack on a part's tensile strength, using 90 and 45 degree rotations of the printed model. Chhetri *et al.* [17] introduced skew along one of the build axes as an attack. Moore *et al.* 2016 [10] modified the amount of extruded source material to compromise the printed object's geometry. Pope and Yampolskiy [24] found that indirect manipulations, such as network command timing modifications and energy supply interruptions, can be potential methods used to sabotage a part. Yampolskiy *et al.* [6] discussed various metal AM process parameters that can be manipulated to sabotage a part's quality, and for the powder bed fusion (PBF) process, the identified parameters included the heat source energy, scanning strategy, layer thickness, and source material properties, such as powder size, form, etc. Yampolskiy *et al.* [8] argued that in the case of metal AM, manipulations of manufacturing parameters can sabotage a part's quality, damage the AM machine, or contaminate the surrounding environment. Slaughter *et al.* [25] showed that for industrial-grade metal 3D printers, a part's quality can be sabotaged indirectly via a compromised *in situ* infrared thermography quality control system.

Two works that also exploit acoustic side channels are directly relevant to our proposal. Chhetri *et al.* [17] presented the first method for the detection of sabotage attacks. The authors used the acoustic side-channel inherent in the FDM process and reported a 77.45% detection rate for object modifications. A recent paper by Bayens *et al.* [18] improved the detection rate by combining the acoustic side-channel measurements with imaging analysis and embedded materials. In the paper, the authors focused on detecting different internal fill structures during the 3D printing process. In the present work, we exploit the acoustic side-channel, which is consistent with Chhetri *et al.* [17] and Bayens *et al.* [18]. However, the proposed algorithm for processing the data is entirely different and enables us to achieve better detectability results.

[1] XCon2013 speakers: http://xcon.xfocus.org/XCon2013/speakers.html

III. Threat Model

Although the central function of any AM process is the 3D printing itself, the process includes many other stages involving other equipment and technologies. Typically, a 3D object's blueprint (in STL, AMF, or 3MF file format) is first stored on a computer. Before printing, the 3D model is "sliced" into individual layers by a software program. Open source software, such as *Slic3r* and *Cura*, is commonly used for desktop 3D printers that employ FDM technology. Certain parameters (e.g., "fill density" and "fill pattern") influence how the source material is actually deposited in an individual layer. The description of these layers can vary greatly between different AM technologies.

The tool path generated in this stage is then transmitted to a 3D printer via USB, SD card or network connection. The tool path is commonly composed of *G-Code* commands, a legacy language for CNC machines. The individual G-Code commands are interpreted by the firmware installed on a 3D printer and translated to electrical signals for individual actuators, such as motors for X/Y/Z movement and filament extrusion or a heater nozzle, etc.

In this workflow, cyber threats arise because each of the 3D model representations can be corrupted. Researchers have shown that the original blueprint file can be corrupted via remote access to the computer [9] or by malware running on a computer [4]. Vulnerabilities in network communications can be exploited to alter print jobs [23]. Moreover, models can be modified or completely substituted by malicious 3D printer firmware [10].

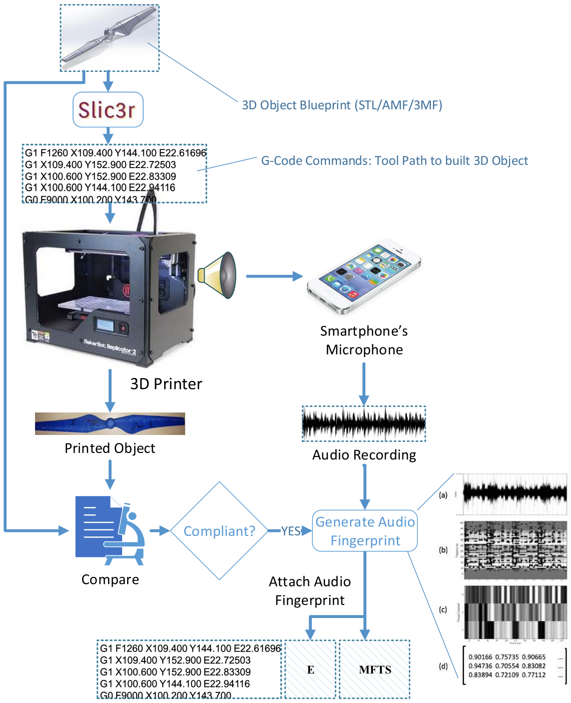
Regardless of the compromised representation, the *cyber-physical* impact depends on the physical change made to the printed object. Researchers have shown that changes to a 3D model alone[2] can prevent its integrability [10], [20], reduce its tensile strength [4], [7], or impact its fatigue life [9]. Particularly in the case of a functional part, such changes can lead to the destruction of cyber-physical systems (CPSs) employing this part, which was shown in the recently conducted *dr0wned* study [9].

IV. Proposed Solution

In this section, we propose a solution for the detection of sabotage attacks that change a 3D printed object's geometry. We describe how the audio fingerprints of the 3D printing process can be generated and verified. We conclude this section by describing how two audio fingerprints can be compared.

A. General Concept: Verification via Audio Side-Channel Fingerprinting

While the protection of each translation stage and representation of a 3D object description is theoretically possible, numerous drawbacks are observed. From an operations point of view, such protection would encompass multiple stages and have negative impacts on the overall performance of the 3D printing process. In the worst case, introducing security measures could interfere with time-critical processes, thereby degrading a manufactured part's mechanical properties. From a security standpoint, the complexity of such a solution would likely be accompanied by new vulnerabilities. Furthermore, if the security mechanisms are integrated into equipment involved in the 3D printing process, then any malicious code that can change the process can also disable or bypass the security mechanisms. Therefore, a verification method that is independent of the manufacturing process equipment must be developed.



```
G1 F1260 X109.400 Y144.100 E22.61696
G1 X109.400 Y152.900 E22.72503
G1 X100.600 Y152.900 E22.83309
G1 X100.600 Y144.100 E22.94116
G0 F9000 X100.200 Y143.700
```

```
G1 F1260 X109.400 Y144.100 E22.61696
G1 X109.400 Y152.900 E22.72503
G1 X100.600 Y152.900 E22.83309
G1 X100.600 Y144.100 E22.94116
G0 F9000 X100.200 Y143.700
```

Fig. 2. Audio fingerprint generation.

Similar to the KCAD approach proposed by Chhetri *et al.* [17], we exploit the fact that in FDM technology, the geometry of a printed object is defined by the movements of four stepper motors (for the X/Y/Z axes and filament extrusion), each of which generates noise with unique characteristics. To detect manipulations, we propose a method based on recording and digitally signing the generated sound by manufacturing a verifiable benign 3D object. Similar to an approach proposed for the detection of hardware Trojans [26], after the sound is recorded, the compliance of the printed 3D object to the blueprint can be validated using destructive methods, and only then can the recorded sound be used as a "fingerprint" of a valid manufacturing process. The workflow for this process is illustrated in Figure 2. The input is either an STL or G-Code file of an object that is produced in a lab environment, and the audio signal is recorded. The fingerprint of the audio signal is calculated (subsection IV-B), encrypted, and concatenated to the G-Code file. When the same 3D object is manufactured again, its validity can be verified by comparing the sound generated during the manufacturing process to the sound of the signed fingerprint.

[2] Changes in manufacturing process as discussed in [6] are beyond the scope of this paper.

In industrial settings, when large runs of the same 3D object are manufactured, fingerprint generation can be performed and verified by the manufacturer. Our approach uses a smartphone for the collection of audio side-channel data; therefore, verification can also become available for home 3D printing users.

Note that the "fingerprint" generation algorithm captures the signal "as is" and does not use audio classifiers, such as in [18], because in the verification phase, the slightest deviations from the "fingerprint" should be detected; however, audio classifiers might compensate for and ignore those minor modifications.

B. Master Audio Fingerprint Generation

Several approaches are available for audio fingerprinting depending on the tasks and challenges involved [27]. The scheme used for this paper is inspired by the idiosyncrasies of the noise emitted by the mechanical components of 3D printers. As shown in [13], all four stepper motors on an FDM 3D printer produce noise with unique characteristics; furthermore, these characteristics distinguish a motor movement's direction, as well as its speed, to some extent. Therefore, we claim that similar motor movements will lead to similar acoustic patterns that can be parameterized and used for comparison to ensure the authenticity of the manufacturing process.

3D printing acoustic patterns are limited and concentrated in specific frequency ranges since they are generated by a fixed combination of mechanical transitions. A common audio fingerprinting approach is to create a summary of an audio recording by parameterizing unique acoustic anchor points in frequency and time.

Accordingly, we propose a method of generating a fingerprint of an FDM 3D printing process via the following steps (see Figure 3). First, we divide the original audio recording into equidistant overlapping time frames and apply a fast Fourier transform (FFT) algorithm on each of these frames. We then use a principal component analysis (PCA) [28] to compress the data by reducing the number of dimensions with minimal loss of information. The use of the PCA enables pattern identification within the signal and a comparison of different signals. The outcome of the PCA transformation step can then be represented as a matrix.

The output of these steps is a text file containing the audio fingerprint, and it can serve as a *master file textual summary* (MFTS) for the printing process and be used to validate future recordings.

Algorithm 1 depicts the pseudo code of the algorithm used to generate the master fingerprint.

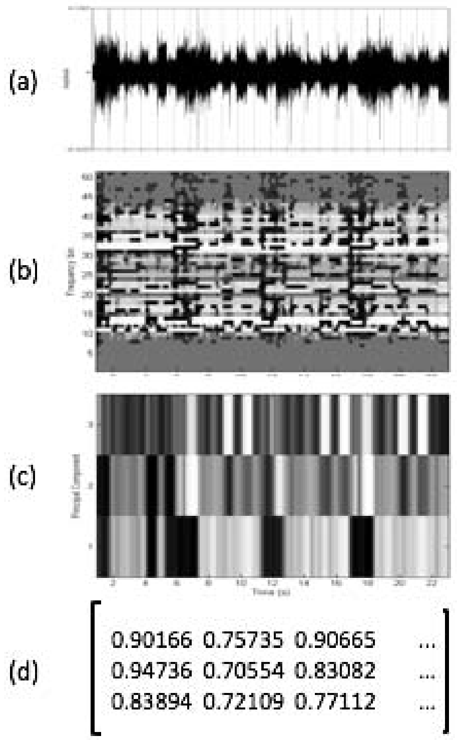


Fig. 3. Audio fingerprint generation: (a) original signal, (b) spectrum after running FFT, (c) gray-scale representation after dimension reduction with the PCA algorithm, and (d) numeric representation of the audio fingerprint.

Algorithm 1 Audio Fingerprint Generation

```
 1: function AudioFingerprintCreation($signal$)
 2:     $TrimByMarkers(signal)$
 3:     $downsample = Resample(signal, 2000)$
 4:     $S = spectrogram(downsample, 0.75, 0.1, 1000, 20)$
 5:     $S = S - mean(S)$
 6:     $covariance = S * S^T$
 7:     $[E] = eigs(covariance, 3)$
 8:     $E = E/norm$
 9:     $MFTS = S * E$
            return $< E, MFTS >$
10: end function
```

Line 2:

First, we bind the audio signal to the section directly related to the manufacturing process. To synchronize the audio recording device with the 3D printer, we insert audible markers at the beginning and end of the 3D printing process. We use the Beep command (M300) and the Dwell command (G4) to signal the boundaries, which allows us to remove irrelevant data after audio recording.

Line 3:

Preliminary experiments indicate that a bandwidth of 1 kHz captures most of the relevant acoustic information for our 3D printer. Therefore, according to the Nyquist rate, the original audio recording can be downsampled to 2 kHz without introducing errors. The downsampling step includes low-pass filtering of all signals with a frequency above 1 kHz. Downsampling reduces the computational complexity of subsequent steps and discards less-informative high-frequency regions.

Line 4:

The spectrogram showing the power density of the downsampled audio record is calculated.[3] We selected

[3] A spectrogram can be created by sequentially calculating the magnitude of the spectrum of overlapping frames of the signal using a fast Fourier transform (FFT) implementation, such as [29].

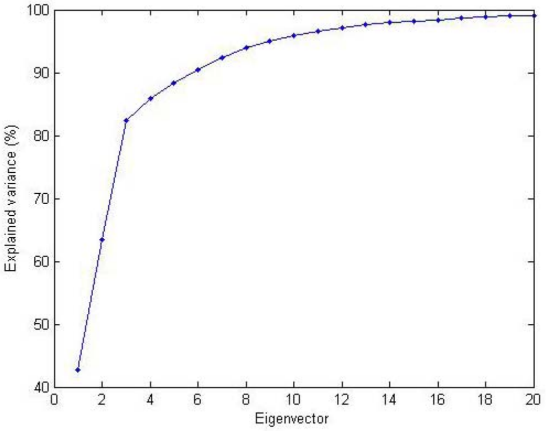


Fig. 4. Eigenvalues of the calculated eigenvectors.

the following spectrogram parameters: The signal is segmented into overlapping frames of 0.75 seconds with a stepping factor of 0.1 second, and the FFT resolution is 20 Hz, resulting in 50 bins that together reach up to 1000 Hz, which is the signal bandwidth. The spectrum of each frame generates a gray-level column along the frequency axis at the corresponding signal time slot. Darker levels represent higher energy densities, and brighter levels represent lower energy densities (Figure 3b).

Lines 5 through 8:

Next, we apply the PCA [28]. The PCA transformation consists of several steps. First, the data are centered by removing the mean spectrum (static component) from each frequency bin in step 5, which facilitates the removal of potential channel mismatches between the current recording and future recordings. Then, the data covariance is calculated (line 6), and it represents a measure of the "spread" of a set of points around their center of mass (mean). Thus, we measure the degree of variation of the dimensions from the mean with respect to each other.

Then, we calculate the eigenvectors of the covariance matrix in step 7 and normalize them in step 8.

The eigenvalues in the PCA indicate the data variance associated with a specific eigenvector. Therefore, the highest eigenvalue indicates the highest variance in the data that was observed in the direction of the associated eigenvector. Accordingly, by using all eigenvectors, we can represent all of the variance found in the data. Figure 4 contains a graph of the variance obtained by adding each eigenvector. In addition to compression, the PCA may help reduce noise by eliminating secondary effects found in less significant eigenvectors.

We empirically identified that three eigenvectors are sufficient to represent the recordings for audio fingerprint generation and perform comparisons for the 3D printers that were tested.

Line 9:

The PCA uses the main eigenvectors of the covariance of the observed data to project it onto an orthogonal low-dimensional subspace. The learned subspace is shown to be closely related to the subspace spanned by the data centroids obtained through unsupervised clustering [30]. These centroids summarize the set of acoustical patterns corresponding to the printer's actions. Therefore, the final step (line 9) of the algorithm is to project the spectrogram matrix onto the three selected eigenvectors, thereby resulting in a stream of vectors with a length of three every 0.1 seconds.

Algorithm output:

The outputs of the algorithm are the MFTS and the three selected eigenvectors calculated for the audio master file.

C. Audio Fingerprint Comparison

Figure 5 illustrates the workflow of the verification process. The input for the process is a signed G-Code file that contains the G-Code commands and the generated signature. This file is submitted to both the 3D printer and the verification mobile device application. The signed G-Code file is then printed on a 3D printer of the same model used to create the fingerprint. In parallel, the audio signal of the printing process is recorded via a special mobile device application, which also receives the signed G-Code file, extracts the signature from the file, decrypts the MFTS and the eigenvectors and compares the recorded audio signal in real time to the MFTS, as described in Algorithm 2.

Algorithm 2 Audio Fingerprint Comparison

```
1: function AUDIOFINGERPRINTCOMPARISON($signal, E, MFTS$)
2:     $TrimByMarkers(signal)$
3:     $downsample = Resample(signal, 2000)$
4:     $S = spectrogram(downsample, 0.75, 0.1, 1000, 20)$
5:     $S = S - mean(S)$
6:     $afterPCA = S * E$
7:     $similarity = cos(afterPCA, MFTS)$
8:     $similarity = smooth(similarity, 3)$
           return $similarity$
9: end function
```

To verify the integrity of the new audio recording, we use a similar algorithm to extract the textual summary of the audio signal and compare it to the MFTS.

The algorithm receives three parameters: the signal of the new 3D printing recording, the MFTS, and the three eigenvectors associated with the MFTS.

Lines 2 through 5:

The initial preparatory operations are identical to those in Algorithm 1.

Line 6:

Next, we calculate the $afterPCA$ value as a projection of the spectrogram on the eigenvectors that were determined in Algorithm 1.

Line 7:

We then use cosine metrics to quantify the similarity between the two vectors. The cosine similarity measures the cosine of the angle between vectors, with a score of 1 corresponding to identical vectors and a score

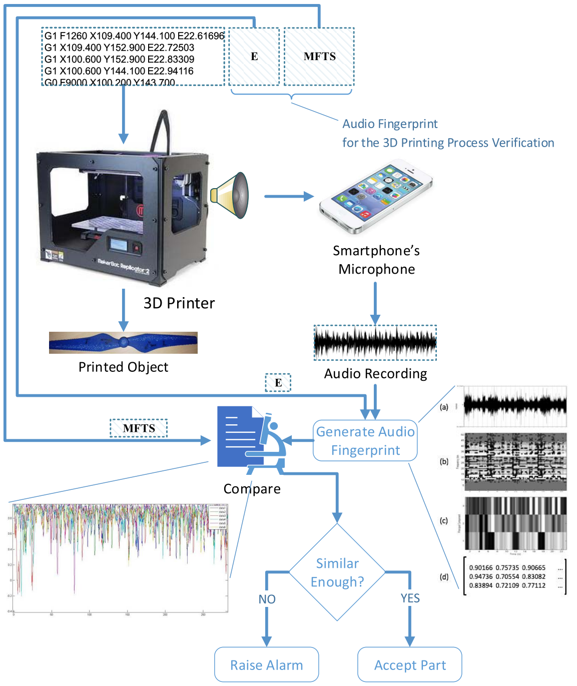


Fig. 5. Audio fingerprint verification.

of 0 corresponding to a lack of correlation between the vectors. Vectors that are correlated in the opposite direction are scored as -1. In this scenario, lower similarity numbers indicate miscorrelations. At the end of this step, we obtain a stream of similarity coefficients.

Line 8:

We apply a moving average filter to smooth out short-term acoustic fluctuations in the similarity stream output and alleviate slight pattern misalignments. Note that the smoothing filter span should match the desired resolution level of the verification processes. For example, a short span is required for the detection of fine printing movement mismatches, although it would likely lead to an increase in false positives, especially in a noisy environment. In our experiments, we set the filter span to 10 (= 1 second).

Alignment is critical for this algorithm since the similarity is calculated by the cosine similarity of frames at the same time offset. Any misalignment could produce negative results.

V. Experimental Evaluation

In this section, we present the experimental evaluation of the proposed detection method. We discuss the setup, the modifications included in the 3D designs that represent changes inflicted by a cyber-attack, and the results (i.e., comparison graphs of the recordings).

A. Experimental Setup

All experiments were performed in a lab environment. We used ordinary PCs to create the G-Code files that were copied to an SD card that was inserted into the 3D printer. The majority of the experiments were performed using a BCN3D Sigma printer that has independent dual extrusion (IDEX) technology, meaning that the 3D printer has two extruder heads that operate independently. The 3D printer runs the "BCN3D Sigma Marlin" firmware and uses FDM technology. The maximum noise level of this 3D printer reaches 58 dBA (A-weighted sound pressure level) as indicated in the technical documentation of the manufacturer.

The software used for slicing the STL files is Cura-BCN3D, a version of the open source Cura software customized for the Sigma 3D printer. The design modifications were created by either changing properties in the Cura-BCD3D software or by modifying the G-Code files. To test the applicability of our solution to other 3D printers, we conducted additional experiments using "MakerBot Replicator Z18" and "Printrbot Plus 1404."

Audio recordings of the 3D printing process were taken using freeware applications on mobile devices. During the 3D printing process, the mobile device was placed adjacent to the 3D printer and recorded the audio in stereo at 44.1 kHz.

B. Experiments Performed

For each modification and disturbance that were tested, we recorded two unmodified objects and at least three modified objects. The modified files presented a decreasing time of disruption in order to determine thresholds at which tampering can be reliably detected. One of the audio recordings of the unmodified 3D object was used to calculate the master audio fingerprint (which consists of the $< E, MFTS >$ tuple generated by Algorithm 1). The other audio recordings were used in Algorithm 2 to verify the detectability of modifications and determine false positive and negative rates.

Although the majority of the experiments were conducted by modifying a specific cube's geometry, we have also validated the experiments on rectangles, pyramids, and more complex geometries (e.g., a propeller).

1) Normal and Abnormal Behavior: To determine the tolerance of the algorithm, we examined the comparison between audio recordings of identical G-Code files and the pre-recorded audio master file. We recorded the audio signals of the printing process of 30 identical cubes in different settings and plotted the results of the comparison algorithm (Algorithm 2).

We tested the robustness of the proposed solution to variations of the following four factors.

Recording Position: The audio for the master signature generation was recorded on the left side of the printer. We tested recordings when the mobile device was placed on the left side of the 3D printer above the print bed (denoted by "Left"), on the right side of the 3D printer above the print bed (denoted by "Right") and at the front of the 3D printer below the print bed (denoted by "Front").

Noise Level: The master file was recorded in a quiet environment (denoted by "No Noise"). In this case, the signal-to-noise ratio (SNR), expressed as the difference between the signal and noise mean energy intensities,

TABLE I

SUMMARY OF THE RECORDING SETTINGS FOR BENIGN 3D PRINTS

NUMBER OF TESTS	RECORDING POSITION	NOISE LEVEL	RECORDING DEVICE	TIME OF RECORDING
12	Left	No Noise	Original	Within two months
2	**Right**	No Noise	Original	Within two months
2	**Front**	No Noise	Original	Within two months
4	Left	**Mild Noise**	Original	Within two months
2	Left	**Loud Noise**	Original	Within two months
4	Left	No Noise	**Different OS and App**	Within two months
4	Left	No Noise	Original	**After 6 months**
30				

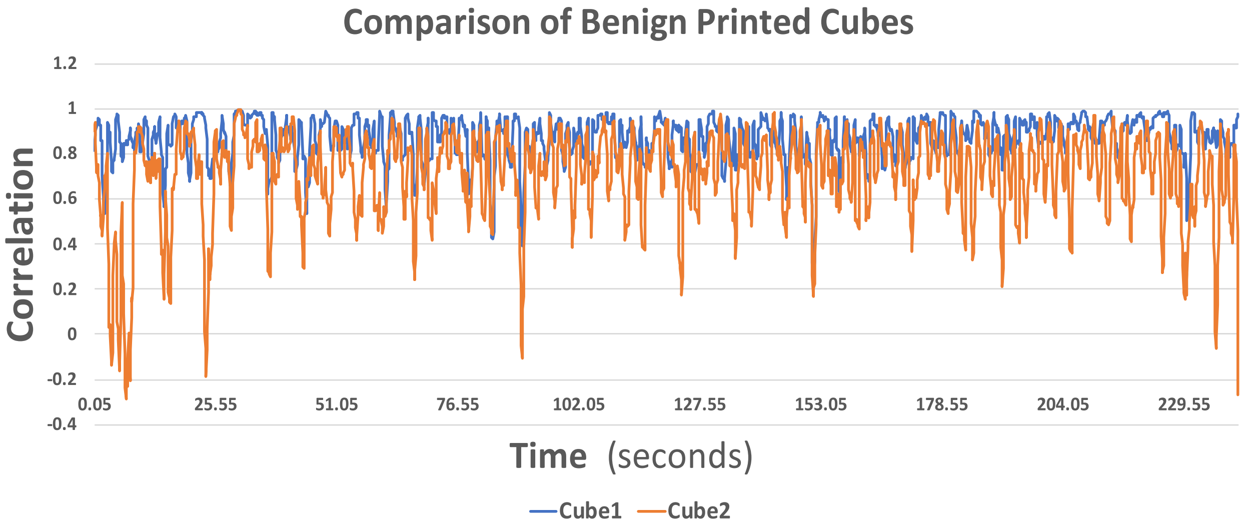


Fig. 6. Comparison of the audio recording of two benign 3D prints (smoothing factor of 10).

was estimated to be 20 dB. This SNR value is clearly compatible with our algorithm. Additionally, we tested cubes that were printed with mild background noise (denoted by "Mild Noise") with an SNR of approximately 0 dB. Finally, we tested cubes that were printed with loud momentary background noises (denoted by "Loud Noise"), and in these cases, the SNR was estimated to be −15 dB.

Recording Devices: We used the same mobile device and mobile application to test the majority of the benign cubes (denoted by "Original"). Several tests were performed with different mobile devices with distinct hardware (microphones) operating under different operating systems (OSs) with different filters and compression methods (denoted by "Different OS & App").

Time of Recording: Most of the tests were performed within two months of recording of the master file (denoted by "Within two months"). Other tests were recorded after six months to test the 3D printer's durability against wear over time (denoted by "After 6 months").

The different recording settings are summarized in Table I and further discussed in subsection V-C2.

The lines in Figure 6 depict the output of Algorithm 2 (a comparison of audio fingerprints) for two identical cubes to the pre-recorded master file. The plot for the first cube (Cube1) is representative of the vast majority of the tested audio recordings for unmodified 3D objects, where the correlation value calculated in Algorithm 2 concentrates approximately 0.8 - 0.9. In contrast, the second cube (Cube2) represents a benign cube that caused a false positive detection. At the beginning of the recording, the correlation level drastically decreased, although the graph syncs back. We observed that this resynchronization always occurs on benign prints and is an indicator of integrity. However, this feature can mask certain attacks, which will be demonstrated later in this section.

Based on these results, we can learn about the normal behavior of the graph when comparing prints of identical objects. For example, we observed that the correlation between the MFTS and each of the audio files in the test set is very high. For a number of the compared files, periodic downward spikes in correlation are observed, and they are typically caused by background noise. However, these spikes are brief, and the correlation quickly returns to higher values.

When observing the correlation graphs of modified objects, a deviation from the expected pattern as well as the exact point when the first deviation occurs can be detected. The line marked as "Bad" in Figure 7 displays the fingerprint output of a recording of a modified cube. The audio recordings of the 3D printing process of the modified cubes lose synchronization exactly when the modification of the G-Code instruction sequence occurs. The "Bad" cube's G-Code contains two dummy moves at layer 20 (out of 40 layers), where the graph loses synchronization roughly in the middle of the recording.

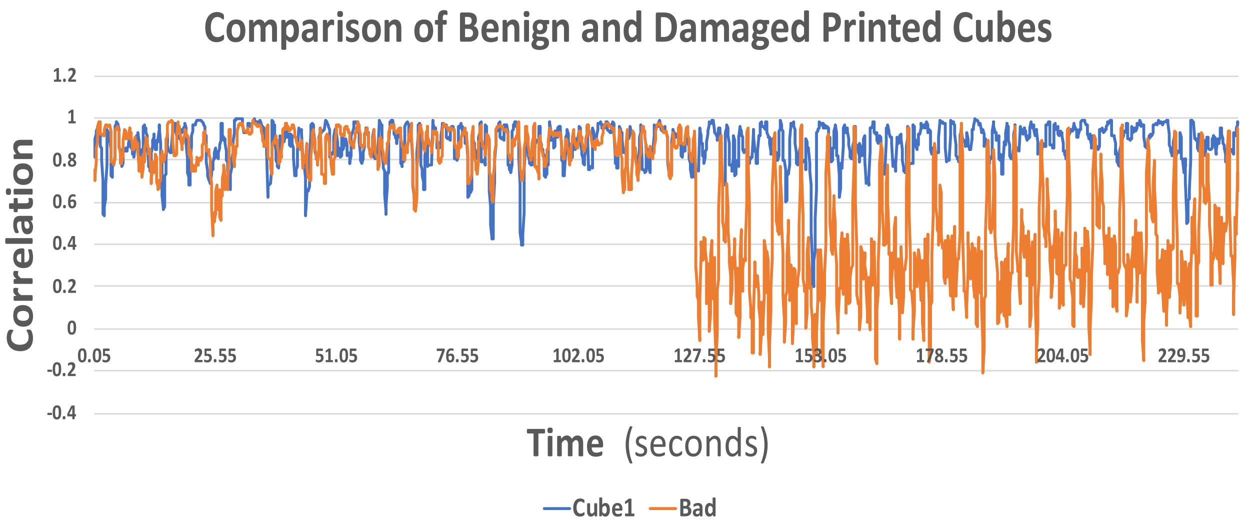


Fig. 7. Comparison of the audio recording of benign and modified 3D prints (smoothing factor of 10).

2) Detection Limits: To determine the detectability thresholds of the proposed method, we systematically search for the minimal malicious modification that can be introduced that is still detected. Such attacks discussed in the literature are modifications of the 3D object geometry [4], [7], [9], [17], 3D object orientation during the printing process [6], [7], and manipulations of the manufacturing process [6]. All these attacks result in changes to the tool path instructions. Attacks involving malicious firmware can deliberately misinterpret G-Code commands [10] and result in actual tool path changes, which can be described (and tested) as changes introduced into the G-Code commands. These modifications will cause corresponding changes in acoustic emanations, such as minor frequency changes. Therefore, to evaluate the method's limitations, we tested changes at the individual G-Code command level. More specifically, we tested the following modifications:

1) Insertion of additional G-Code commands;
2) Deletion of G-Code commands included in the tool path of a benign 3D object;
3) Modification of parameters for an individual movement command along one axis;
4) Modification of the extruder's speed;
5) Reordering of G-Code commands.

To analyze the detection thresholds, we performed controlled tests on 20 second long sections of audio recordings. We used a cube in our experiments due to its repetitive geometry, and each section translates to exactly four layers in the cube. Each section was marked with the audio marker, and the modification was inserted in the third layer, *i.e.,* in the second half of the recording.

For every modification type, we tested decreasing levels of deviations and validated whether or not such deviations can be detected with the proposed approach.

The comparison was performed with a smoothing factor of three. Although a lower factor can improve the detection resolution, it would increase the false positive rate.

TABLE II

INSERTION OF TWO G0 MOVES

Original Code	Insertion of Commands
G1 F1500 X108.370 Y144.239 E36.097 G0 F9000 X109.400 Y144.100	G1 F1500 X108.370 Y144.239 E36.097 **G0 F9000 X102.766 Y144.239** **G0 F9000 X103.745 Y152.759** G0 F9000 X109.400 Y144.100

C. Experimental Results

In this subsection we outline the experiments that were performed and focus on detecting abnormal behavior on the similarity graph and narrowing down the lower bound parameters for detecting cyber-physical attacks.

Note that the graphs in this section are plotted with various smoothing factors for visual clarity. The verification algorithm was calculated with data using a smoothing factor of three.

1) Results of Atomic Modifications: The experimental evaluation of the ability to detect atomic modifications is as follows.

a) ***Insertion of Commands****:* We inserted additional G0 commands into the manipulated G-Code files. The G0 command translates to an extruder movement towards the specified X and Y coordinates without extruding the filament. Table II shows the original and modified G-Code commands of one file (two additional G0 commands were inserted). Figure 8 shows the similarity graph comparing the 3D printing process of three modified G-Code files and the audio master file. The modified files contained four, two, and a single inserted G0 command. The addition of G0 commands desynchronizes the audio, and the degree of similarity consequently degrades dramatically immediately following the execution of the inserted commands.

b) ***Deletion of Commands****:* The G1 command moves the extruder to a specified (X, Y) coordinate while extruding the filament. Changes to the 3D object geometry (both internal and external) will likely involve modification of the G1 commands. To validate the detectability of such changes, we deleted G1 commands from the G-Code files. Table III shows both the

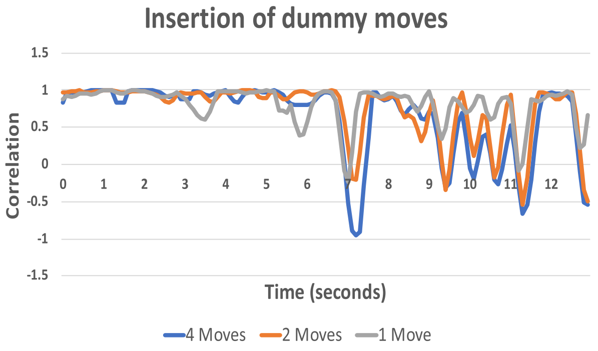


Fig. 8. Comparison of the audio recording for the master cube versus three modified cubes (consisting of the insertion of G0 moves).

TABLE III

DELETION OF TWO G1 PRINT MOVES

Original Code	Deletion of Commands
G1 F1260 X109.400 Y144.100 E52.26532	G1 F1260 X109.400 Y144.100 E52.26532
G1 X109.400 Y152.900 E52.37338	G1 X109.400 Y152.900 E52.37338
G1 X100.600 Y152.900 E52.48145	G0 F9000 X100.200 Y143.700
G1 X100.600 Y144.100 E52.58952	
G0 F9000 X100.200 Y143.700	

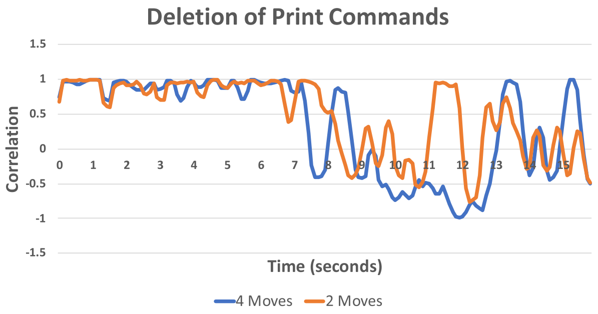


Fig. 9. Comparison of the audio recording of the master cube and two modified cubes (the modification consisted of the deletion of G1 print moves).

original and modified G-Code commands on one file. Figure 9 shows the similarity graph comparing the 3D printing process of the unmodified file and the two modified files, which had four and two removed G1 commands, respectively. Even the deletion of a single G1 command might disturb synchronization, which is reflected by the dramatic degradation of the degree of similarity immediately following the removal of the G-Code command.

*c) **Modification of Movement Length on the Axis:*** We tested several modifications of movements along the axes, including extending and shortening the length of the move. In both cases, we successively reduced the length of the deviation from the original command and observed the impact on the similarity plot.

Detection depends on the time delays introduced into the printing process. Even a minimal deviation length can be detected if it is printed at the right feed rate. The feed rate parameter also influences the speed of the move; thus, shorter move lengths require slower movement speeds for detection. In this experiment, a feed rate of 1260 is used, and the minimum change that still disturbed the synchronization at our smoothing factor is a modification of 1 cm in length on a single G1 print command. This resulting break in synchronization is similar to that achieved by inserting or deleting a G1 print command (see Figure 8). Modifications of 0.5 cm and 0.2 cm also break the synchronization, although in these cases, the degree of similarity is not reduced as drastically.

TABLE IV

EXTENDING A SINGLE G1 PRINT MOVE IN G-CODE

Original Code	Extend a command on the Y axis
G1 F1260 X109.400 Y144.100 E47.32392	G1 F1260 X109.400 Y144.100 E47.32392
G1 X109.400 Y152.900 E47.43199	G1 X109.400 Y152.900 E47.43199
G1 X100.600 Y152.900 E47.54006	G1 X100.600 Y152.900 E47.54006
G1 X100.600 **Y144.100** E47.64813	G1 X100.600 **Y139.100** E47.64813
G0 F9000 X100.200 Y143.700	G0 F9000 X100.200 Y143.700

TABLE V

MODIFICATION OF THE FEED RATE OF TWO PRINT COMMANDS

Original Code	Extend a command on the Y axis
G1 F1260 X109.400 Y144.100 E47.32392	G1 F1260 X109.400 Y144.100 E47.32392
G1 X109.400 Y152.900 E47.43199	G1 X109.400 Y152.900 E47.43199
G1 X100.600 Y152.900 E47.54006	G1 X100.600 Y152.900 E47.54006
G1 X100.600 **Y144.100** E47.64813	G1 X100.600 **Y139.100** E47.64813
G0 F9000 X100.200 Y143.700	G0 F9000 X100.200 Y143.700

Table IV shows the original and modified G-Code commands; in this case, the modification extends the Y-axis movement by 0.5 cm. As a result of this modification, the correlation graph loses synchronization at the point of the change. The experiments were performed for all three axes and present uniform detectability thresholds.

*d) **Modification of Extruder Speed:*** The amount of filament deposited during a movement is a function of the speed of the nozzle movement and the speed of the filament extrusion motor (both controlled by G-Code command G1). Modifying the feed rate parameter of the G1 commands changes the speed of the executed move.

In this case, determining the minimal change needed to break synchronization is more difficult because two factors are involved: the length of the move and the original feed rate of the command. We found that we could break synchronization by slowing down the speed of two G1 print commands (a 1 cm move length).

Table V shows the original and modified extrusion feed rate parameters of the G1 G-Code command. As a result of this modification, the correlation graph loses synchronization at the point of the change.

*e) **Reordering of G-Code commands:*** Reordering G-Code commands do not modify the geometry of the object but might affect the quality of the object. However, reordering a few commands does not appear to hurt the overall synchronization. Table VI shows reordering of G-Code commands of a single layer in the 3D printed cube. The result of this change produces a momentary decrease in the correlation graph. Hence, since the overall timing has not changed, the line resyncs afterwards. When we examine the effect of reordering on the comparison graph of the entire cube (Figure 10), a noticeable disturbance is observed near the beginning of the file, although no disturbances are observed afterward. Therefore, we conclude that the reordering of commands will

TABLE VI
REORDERING THREE G1 PRINT COMMANDS IN G-CODE

Original Code	Reorder of Commands
G1 F1260 X109.400 Y144.100 E22.61696	**G1 F1260 X100.600 Y152.900 E22.83309**
G1 X109.400 Y152.900 E22.72503	**G1 X109.400 Y152.900 E22.72503**
G1 X100.600 Y152.900 E22.83309	**G1 X109.400 Y144.100 E22.61696**
G1 X100.600 Y144.100 E22.94116	G1 X100.600 Y144.100 E22.94116
G0 F9000 X100.200 Y143.700	G0 F9000 X100.200 Y143.700

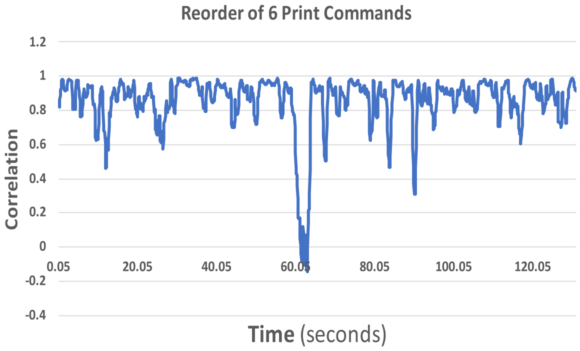


Fig. 10. Comparison of the audio recording of the master cube and a modified cube (the modification consists of reordering six G1 print commands and a smoothing factor of 10).

be noticeable but difficult to distinguish from disturbances introduced by background noise.

2) Testing Disturbances: Several factors can influence the signal of an audio recording. To test the resilience of the proposed solution to various factors, such as the recording device type, microphone position, and background noise, we performed several audio recordings while introducing disturbances. A description of the tested disturbances follows.

a) Different Recording Devices: In production, the audio signal would be recorded by different applications on different mobile devices. Thus, we compared several mobile applications across different mobile devices and obtained the same results.

b) Recording Positions: The recording position (the location of the microphone in relation to the 3D printer) varied during certain experiments. Although the audio master file was recorded with the microphone positioned on the left side of the 3D printer above the extruder, additional audio recordings were taken with the microphones on the right side of the printer or in the front at approximately 20 cm below the extruder head.

c) Background Noise: The recordings were performed in a lab environment, with some performed at night in a quiet environment and others conducted in the daytime with mild background noise. The effects introduced by the noisy environment appear as short negative peaks in the similarity graph. These drops are limited to the duration of the noise, and the background noise does not affect the synchronization of audio recordings. In an extremely loud environment with permanent background noise, this behavior might cause false positives. Figure 11 depicts the effects of loud momentary noises during the recording process. The environmental noise causes the large negative peaks in synchronization and the rapid recovery.

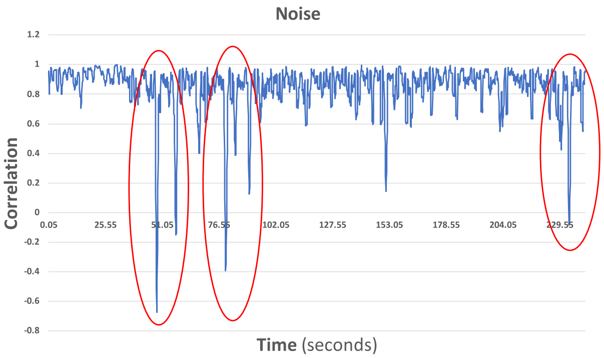


Fig. 11. Comparison of the audio recording of the master cube and an identical cube that was recorded with background noises (smoothing factor of five).

d) Different Object Geometry: We tested different object geometries to ensure that the results can be recreated on any geometry. The geometries used are as follows:

1) Cube - the main shape that was used during our experiments (Figure 12a);
2) Rectangle - different sizes of rectangles were tested to ensure that the size does not influence the results (Figure 12b);
3) Pyramid - varying layer sizes were tested to determine whether modifications in smaller layers could be distinguished (Figure 12c);
4) Nut - a circular shape was used to validate that the proposed approach can detect circular movements in addition to long linear movements (Figure 12d);
5) Propeller - the algorithm was tested in a real-world attack scenario in which a drone's propeller was modified; this scenario represented one of the motivations for this study [9] (Figure 12e).

e) Different 3D Printers: The majority of the experiments were performed using the "BCN3D Sigma Marlin", although we also verified the results on the "MakerBot Replicator Z18" and "Printrbot Plus 1404." All of the 3D printers operate with the FDM technology on plastic. The recordings were performed by different people with different mobile devices and in different rooms (acoustic settings).

In future work, we will further explore the case of intra-class variance produced by different 3D printers of the same brand and model running the same G-Code file. From the preliminary tests performed by our group in this area, we observed that a linear drift in time might occur, although it can be overcome by adding a calibration step to our comparison algorithm. In future work, we plan to generalize the calibration step to enable comparisons of acoustic signatures between 3D printers of the same brand and model and 3D printers of different brands and models.

D. Algorithm Limitations

Pattern similarity is calculated on a frame-by-frame basis; therefore, the algorithm relies on time synchronization. Modifications that cause a momentary mismatch in time

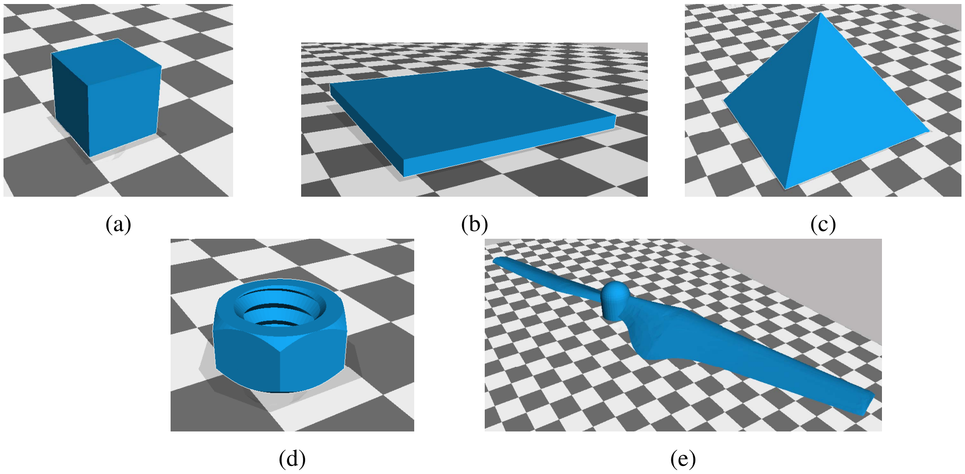


Fig. 12. Images of the geometries tested in both benign and malicious scenarios. (a) Cube. (b) Rectangle. (c) Pyramid. (d) Nut. (e) Propeller.

TABLE VII
REPLACE 2 G1 COMMANDS WITH G0 COMMANDS IN G-CODE

Original Code	Change Print Commands
G1 F1260 X109.400 Y144.100 E22.61696	G1 F1260 X109.400 Y144.100 E22.61696
G1 X109.400 Y152.900 E22.72503	G1 X109.400 Y152.900 E22.72503
G1 X100.600 Y152.900 E22.83309	**G0** F1260 X100.600 Y152.900
G1 X100.600 Y144.100 E22.94116	**G0** X100.600 Y144.100
G0 F9000 X100.200 Y143.700	G0 F9000 X100.200 Y143.700

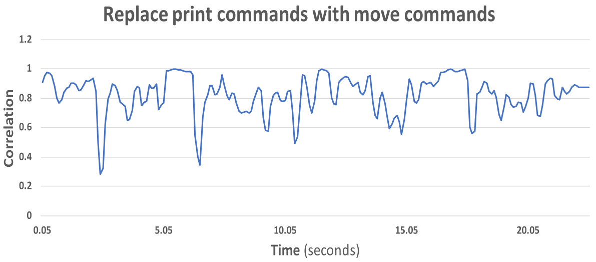


Fig. 13. Comparison of the audio recording of the master cube versus a modified cube. The modification consists of replacing two G1 commands with G0 commands.

synchronization but do not break the overall synchronization might therefore be interpreted as false positives. The main limitation we discovered is in the detection of command replacements of identical length, *e.g.,* replacing G1 print commands with dummy G0 move commands with the same feed rate. The move still occurs but without filament extrusion. This process does not affect the subsequent synchronization; therefore, the audio difference is momentary. We tested the replacement of two G1 print moves with G0 moves with the same feed rate (Table VII). The correlation of the audio signal is still very high and does not trigger an alert. The audio recording of this modification is highly correlated with the audio master file as shown in Figure 13. Thus, the algorithm will not detect this modification.

VI. Quality of the Proposed Method

In this section, we present the modification indicator method and its results. Then, we discuss the strengths of the proposed detection method.

A. Modification Indicator

To detect modifications, we searched for significant changes in the mean value of the signal. To eliminate large but brief dips in the correlation graph, which are frequently caused by background noise, we applied a large smoothing factor to the similarity graph. We calculated the mean value for time frames of 5 seconds and searched for locations in which an overall decrease on the correlation axis occurs (of at least 0.37 points) for four consecutive time frames. Out of 30 identical printed cubes, one false negative alert was observed for a benign cube, and it was due to artificially introduced loud background noise.

Table VIII summarizes the results of the modification detection experiments. The comparison algorithm primarily relies on timing; therefore, we experimentally estimated the threshold in seconds of minimal introduced deviation that would be detected with the parameters listed above. The “Detectable Threshold” column in the table represents the minimal deviation (in seconds) that was reliably detected for each modification, i.e., **all** modified samples that had deviations of the specified time and larger were correctly classified as attacks. The “Undetected Threshold” column represents the maximal deviation time under which the modification could not be detected at least once, i.e., **at least one** of the modified samples was incorrectly classified as benign.

Based on our experiments, we conclude that under the presented experimental environment and assumptions, any deviation (except reordering) of **one second** would be classified as an attack and result in an alert. In contrast, a reordering modification should last more than 2.66 seconds to be detected.

B. Discussion of Testing Disturbances

One of the greatest challenges in audio processing is the issue of channel mismatch. Several factors might introduce noise into the recordings and consequently distort the extracted features in the modeling signature and test templates. The most dominant factors are background noise and variations among the microphones, filters, and compression methods of different recording devices. Moreover, temporal distortions, specifically reverberation, can arise due to different acoustic

TABLE VIII

SUMMARY OF THE DETECTION THRESHOLDS ON VARIOUS MODIFICATIONS

Detectability Thresholds		
Modification	**Detectable Threshold**	**Undetectable Threshold**
Insertion of Commands	0.58 sec	0.34 sec
Deletion of Commands	0.417 sec	0.355 sec
Length Change of Commands	0.58 sec	0.34 sec
Print Feedrate Change	0.834 sec	0.8 sec
Command Reordering	2.66 sec	1.15 sec

paths between the 3D printer and the recording device. In this case, the recording device captures delayed copies of the sound, which leads to synchronization difficulties. During the tests, we observed that certain settings result in minor channel mismatches and affect the amount of noise that is added to the correlation graph. However, we selected the modification indicator so that it is more tolerable for benign prints, i.e., the algorithm will overcome noisy channels (up to a certain amount of noise). Out of the disturbances that were tested in subsection V-C2, the following disturbances caused channel mismatches:

1) Different recording positions that impact the distance of the recording device from the printing head;
2) Different recording devices with different hardware and software;
3) Background noise;
4) Elapsed time from the recording of the master file.

Tests that involved different geometries or the algorithm applied to other 3D printers did not cause channel mismatch. Figure 14 demonstrates the "noise" added to the correlation graph. The sample was recorded six months after the recording of the master sample and by using a different mobile device. The correlation graph has more negative peaks than the correlation graph presented in Figure 6. If we construct the mean value graph via the method described in subsection VI-A, then the summarized correlation graph clearly does not pass the detection threshold (i.e., does not contain drops of 0.37 points). Figure 15 shows the mean value graph of the correlated signature of the tested disturbance sample. The graph also contains the mean value of a modified sample that contains an attack to clarify how the modification indicator is passed in attack samples.

C. Detectability of Real Sabotage Attacks

Each of the introduced and tested modifications to G-Code commands (e.g., insertion, deletion, reordering, etc.) can be considered an *atomic* (or a *minimally* possible) modification. Due to their minimalistic nature, a single command modification is unlikely to cause a full-fledged sabotage attack capable of impairing the mechanical properties of a manufactured part. Therefore, we also tested the ability of the proposed approach to detect real sabotage attacks that have been presented in the research literature. We simulated the sabotage attacks according to details provided in the research literature, and the results of this verification are summarized in Table IX.

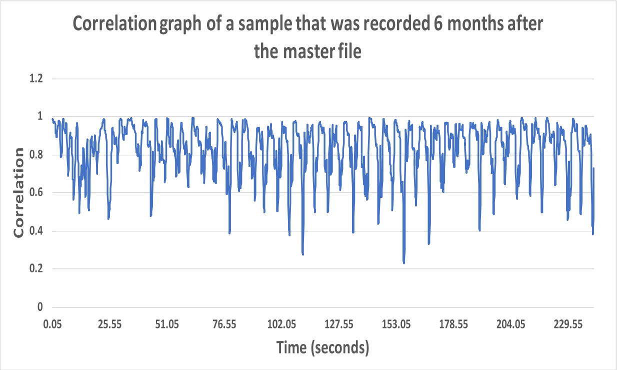


Fig. 14. "Noisy" correlation graph of a sample that was recorded six months after the master file.

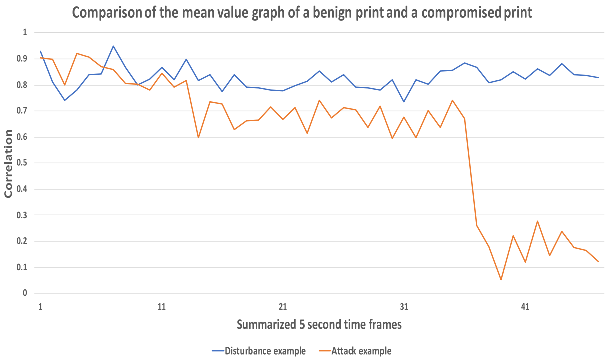


Fig. 15. Comparison of the mean value of the correlation between a benign sample with high disturbance and a malicious sample.

TABLE IX

DETECTABILITY OF SABOTAGE ATTACKS BY THE PROPOSED METHOD

SABOTAGE ATTACK	PROPOSED BY	DETECTED?
Gap/Void	[4], [9]	✓
Contaminant Material	[7]	*N/A*
Different Layer Thickness	[31]	✓
Scale of the Printed Object	[20]	✓
Amount of Extruded Filament	[10]	X
Z-Orientation	[6], [7]	✓
Orientation in X-Y Plane	[7]	✓
Temperature of Extruded Filament	[20]	X
Fill Pattern modification	[18]	✓

✓- Fully detects
X - Does not detect
N/A - Not applicable

Sturm *et al.* [4] and Belikovetsky *et al.* [9] used artificially inserted gaps to reduce a part's tensile strength and fatigue life, respectively. The insertion of a void in the STL file typically leads to numerous modifications of G-Code command sequences generated by a slicer for the affected layers. Our algorithm demonstrated the ability to reliably detect such attacks during the tests.

A sabotage attack that scales up or down a printed object along one or more dimensions will ultimately impact its size. In the case of a functional part, such an attack can impact its ability to fit into the target system and will be reflected in the G-Code commands, and the resulting audio signal will not match the pre-generated signature.

If the layer thickness is changed for all (or many) layers [31], the time deviations will accumulate, resulting in reliable attack detection because this accumulation affects the G-Code of the entire layer and breaks synchronization.

In general cases of object scaling [20], when scaled in the z direction, the number of layers generated during model slicing can change. If the scale affects the x or y directions, each straight or diagonal move involving these dimensions will require a different amount of time or will be performed at a different speed.

All of these factors lead to reliable attack detection.

Yampolskiy *et al.* [6] argued that because of the anisotropy inherent in 3D printed parts, changes of the build orientation can be used as a sabotage attack, which was empirically demonstrated by Zeltmann *et al.* [7]. Our approach reliably detects changes in the orientation of printed objects, including the orientation in the z direction and in the X-Y plane because significant differences in the G-Code command sequences (and resulting motor movements) are caused by the change in orientation.

These findings are consistent with those for object substitution attacks as proposed by Moore *et al.* [10].

Our detection method relies on synchronization and an accurate 3D printing process; thus, changes to infill patterns, such as those presented in [18], are noticeable and detected at the first occurrence.

Changes in the extruded filament temperature, as demonstrated by Hang [20], cannot be detected with the proposed method. Changes in temperature do not impact the movement of the mechanical parts and thus have no effect on the sound generated by the stepper motors during 3D printing.

Changes in the amount of filament that is extruded might not be detected by the proposed algorithm if the movement's speed along the X-Y-Z coordinates remains the same. This issue is covered in subsection V-D.

D. Signature Uniqueness

Using the proposed algorithm, an adversary cannot change the G-Code file to produce the same digital signature. The sound generated by each G-Code command varies based on the frequency and amplitude according to the speed, direction, and extruded material of the command [17]; therefore, an identical sound signal cannot be generated by printing a different 3D object. Moreover, the signing algorithm does not lose critical signal information during data compression. The FFT algorithm, which is used as the first step of compression, is a reversible algorithm that requires frequency, amplitude, and phase data to reconstruct the original signal. In our algorithm, we save the frequencies and amplitudes of each frame, and because the frames overlap with a step of 0.1 second, they account for the phase information. The PCA calculation, which is the second step of data compression, can be adjusted to account for lost information. More than 95% of the original signal can be represented by calculating 10 eigenvectors instead of three as shown in Figure 4, thereby resulting only in an increase of the signature length.

E. Comparisons With Other Acoustic Detection Methods

The use of an acoustic side-channel as a detection mechanism for cyber-attacks was discussed recently in [17] and [18]. This detection method is highly relevant in the case of AM technology because it enables the detection of defects at the very last stage of the manufacturing process. By using a side-channel technology, the detection method is not exposed to the same cyber threats as the manufacturing process itself, and it can reside on an external device and a different network. In this work, we built upon the initial results of previous work and created algorithms that achieve higher accuracy and detection resilience. Compared with [17], we do not try to separate each G-Code command and train a machine learning model to identify the command but rather identify the continuity of the signal and "fingerprint" it. This approach can overcome random noise that can be inserted into the manufacturing process and enable better detection. The accuracy of detection in [17] was 77.45%, whereas the detection reported here is complete and only depends on the size of the change. If Chhetri et al.'s metric is applied to the design used in this paper (a small cube), a change of 22% of the cube's G-Code commands is equivalent to approximately 330 G-Code commands, and accounting for the speed of the print indicates that approximately 60 seconds of printing time might be modified. Thus, theoretically, a well-crafted adversary cube might deviate from the original design by up to 22%, which is equivalent to approximately 60 seconds of printing time. Our approach demonstrated the ability to detect deviations that are greater than one second. Moreover, when performing our experiments and constructing the "fingerprint" generation algorithm, we tried to avoid the use of sound classifier algorithms such as those used in [18] because although they are sufficient for detecting large changes, such as a different fill pattern, they compensate for and ignore small changes that affect only several G-Code commands. The approach proposed by Bayens *et al.* [18] detects changes to the fill patterns that might influence the integrity of the object. In their results, they changed the fill patterns of several layers and could detect honeycomb fill changes within 270 seconds, or 60% of the printing time, and rectilinear fill changes within 180 seconds, or 40% of the printing time. In their experiments, the first three layers that take up to 90 seconds to print remained unmodified. Thus, Bayens et al.'s algorithm can detect changes starting at 90 seconds (180 - 90) up to 180 seconds (270 - 90). In our paper, we showed that we can detect all deviations (not just fill pattern-related modifications) that are greater than one second.

VII. Conclusions

In this paper, we presented a side-channel verification method to ensure the integrity of 3D printed objects. We introduced two algorithms: one for creating the digital audio signature and another for verifying the 3D printing process in real time. The paper includes the results of experiments performed to detect each *atomic* modification, such as insertion, deletion, reordering, and modifying parameters of a single G-Code command. Hence, the proposed detection method is highly efficient in detecting *cyber-physical* attacks that aim to modify the object's geometry or the printing process timing.

References

[1] T. Kellner, "The FAA cleared the first 3D printed part to fly in a commercial jet engine from GE," Gen. Electr., New York, NY, USA, Tech. Rep., 2015. [Online]. Available: https://www.ge.com/reports/post/116402870270/the-faa-cleared-the-first-3d-printed-part-to-fly-2/

[2] T. Wohlers, *Wohlers Report 2017 3D Printing and Additive Manufacturing State of the Industry Annual Worldwide Progress Report*. Fort Collins, CO, USA: Wohlers Associates, 2017. [Online]. Available: www.wohlersassociates.com

[3] A. Müller and S. Karevska, "How will 3D printing make your company the strongest link in the value chain?" Ernst & Young, London, U.K., Tech. Rep., 2016. [Online]. Available: http://www.ey.com/Publication/vwLUAssets/ey-global-3d-printing-report-2016-full-report/$FILE/ey-global-3d-printing-report-2016-full-report.pdf

[4] L. Sturm, C. Williams, J. Camelio, J. White, and R. Parker, "Cyber-physical vunerabilities in additive manufacturing systems," *Context*, vol. 7, no. 7, p. 8, 2014.

[5] M. Yampolskiy, T. R. Andel, J. T. McDonald, W. B. Glisson, and A. Yasinsac, "Towards security of additive layer manufacturing," presented at the 30th Annu. Comput. Secur. Appl. Conf. (ACSAC), 2014. [Online]. Available: http://www.soc.southalabama.edu/faculty/yampolskiy/ Publications/yampolskiy2014towards.pdf

[6] M. Yampolskiy, L. Schutzle, U. Vaidya, and A. Yasinsac, "Security challenges of additive manufacturing with metals and alloys," in *Critical Infrastructure Protection IX*. Cham, Switzerland: Springer, 2015, pp. 169–183.

[7] S. E. Zeltmann, N. Gupta, N. G. Tsoutsos, M. Maniatakos, J. Rajendran, and R. Karri, "Manufacturing and security challenges in 3D printing," *J. Minerals*, vol. 68, no. 7, pp. 1872–1881, 2016.

[8] M. Yampolskiy, A. Skjellum, M. Kretzschmar, R. A. Overfelt, K. R. Sloan, and A. Yasinsac, "Using 3D printers as weapons," *Int. J. Crit. Infrastruct. Protection*, vol. 14, pp. 58–71, Sep. 2016.

[9] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici, "dr0wne—Cyber-physical attack with additive manufacturing," in *Proc. 11th USENIX Workshop Offensive Technol. (WOOT)*, 2017, pp. 1–15.

[10] S. B. Moore, W. B. Glisson, and M. Yampolskiy, "Implications of malicious 3D printer firmware," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017, pp. 6089–6098.

[11] M. Yampolskiy, T. R. Andel, J. T. McDonald, W. B. Glisson, and A. Yasinsac, "Intellectual property protection in additive layer manufacturing: Requirements for secure outsourcing," in *Proc. 4th Program Protection Reverse Eng. Workshop*, 2014, Art. no. 7.

[12] A. Brown, M. Yampolskiy, J. Gatlin, and T. Andel, "Legal aspects of protecting intellectual property in additive manufacturing," in *Proc. Int. Conf. Crit. Infrastruct. Protection*, 2016, pp. 63–79.

[13] M. A. Faruque, S. R. Chhetri, A. Canedo, and J. Wan, "Acoustic side-channel attacks on additive manufacturing systems," in *Proc. ACM/IEEE Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2016, Art. no. 19. [Online]. Available: http://aicps.eng.uci.edu/papers/3-d-printer-security-alfaruque.pdf

[14] J. Blackman, "The 1st amendment, 2nd amendment, and 3D printed guns," *Tennessee Law Rev.*, vol. 81, p. 479, 2013.

[15] K. F. McMullen, "Worlds collide when 3D printers reach the public: Modeling a digital gun control law after the digital millenium copyright act," *Michigan State Law Rev.*, p. 187, 2014.

[16] J. J. Johnson, "Print, lock, and load: 3-D printers, creation of guns, and the potential threat to fourth amendment rights," *Univ. Illinois J. Law, Technol. Policy*, p. 337, 2013.

[17] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, "KCAD: Kinetic cyber-attack detection method for cyber-physical additive manufacturing systems," in *Proc. 35th Int. Conf. Comput.-Aided Design*, 2016, Art. no. 74.

[18] C. Bayens, T. Le, L. Garcia, R. Beyah, M. Javanmard, and S. Zonouz, "See no evil, hear no evil, feel no evil, print no evil? Malicious fill patterns detection in additive manufacturing," in *Proc. 26th USENIX Secur. Symp.*, 2017, pp. 1181–1198.

[19] M. Yampolskiy *et al.*, "Security of additive manufacturing: Attack taxonomy and survey," *Additive Manuf.*, vol. 21, pp. 431–457, May 2018.

[20] X. Z. Hang and C. Xiao, "Security attack to 3D printing," in *Proc. xFocus Inf. Secur. Conf.*, 2013. [Online]. Available: http://www.claudxiao.net/Attack3DPrinting-Claud-en.pdf

[21] H. Turner, J. White, J. A. Camelio, C. Williams, B. Amos, and R. Parker, "Bad parts: Are our manufacturing systems at risk of silent cyberattacks?" *IEEE Security Privacy*, vol. 13, no. 3, pp. 40–47, May/Jun. 2015.

[22] S. Moore, P. Armstrong, T. McDonald, and M. Yampolskiy, "Vulnerability analysis of desktop 3D printer software," in *Proc. Resilience Week (RWS)*, Aug. 2016, pp. 46–51.

[23] Q. Do, B. Martini, and K.-K. R. Choo, "A data exfiltration and remote exploitation attack on consumer 3D printers," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 10, pp. 2174–2186, Oct. 2016.

[24] G. Pope and M. Yampolskiy. (2016). "A hazard analysis technique for additive manufacturing." [Online]. Available: https://arxiv.org/abs/1706.00497

[25] A. Slaughter, M. Yampolskiy, M. Matthews, W. E. King, G. Guss, and Y. Elovici, "How to ensure bad quality in metal additive manufacturing: *In-Situ* infrared thermography from the security perspective," in *Proc. 12th Int. Conf. Availability, Rel. Secur.*, 2017, Art. no. 78.

[26] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 296–310.

[27] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 41, no. 3, pp. 271–284, 2005.

[28] F. R. S. K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philosoph. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

[29] M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, May 1998, pp. 1381–1384.

[30] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 29.

[31] M. Vaezi and C. K. Chua, "Effects of layer thickness and binder saturation level parameters on 3D printing process," *Int. J. Adv. Manuf. Technol.*, vol. 53, nos. 1–4, pp. 275–284, 2011.

Sofia Belikovetsky received the B.Sc. degree in computer science from Tel Aviv University and the M.Sc. degree in computer science from the Interdisciplinary Center Herzliya. She is currently pursuing the Ph.D. degree with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev. The topic of her Ph.D. dissertation is cyber threats and opportunities in additive manufacturing, where she explores offensive methods for attacking the manufacturing processes and ways to defend against them. She is currently a Cyber Security Researcher with a military and government background, where she has held various positions, starting from a software engineer, a pen-tester, a team leader, and the chief architect of her department.

Yosef A. Solewicz received the B.Sc. degree in electrical engineering from the Technion–Israel Institute of Technology, Haifa, Israel, in 1990, the M.Sc. degree in electrical engineering from Catholic University, Rio de Janeiro, Brazil, in 1993, and the Ph.D. degree in computer science from Bar-Ilan University, Ramat-Gan, Israel, in 2006. He is currently with the Israeli Police and also with the Cyber Security Research Center, Ben-Gurion University, actively researching audio, speech, and language processing.

Mark Yampolskiy received the Ph.D. degree in computer science from the Ludwig-Maximilians University of Munich, Germany. He is currently an Assistant Professor with the School of Computing, University of South Alabama. Since his post-doctoral appointment at Vanderbilt University, he performs research on security of cyber-physical systems. He was among the scientists who pioneered Security of Additive Manufacturing (AM, also known as 3-D Printing) around 2014. AM Security remains his major research interest ever since. He has numerous publications in the field, ranging from attacks on or with AM up to novel approaches for the detection of such attacks.

Jinghui Toh was a Researcher with iTrust, the Centre for Cyber Security Research, Singapore University of Technology and Design. He is currently a Cyber Security Consultant focusing on cyber security for cyber physical systems. His research interests are cyber security and privacy for Internet of Things, cyber physical systems, and applications to additive manufacturing.

Yuval Elovici received the B.Sc. and M.Sc. degrees in computer and electrical engineering from the Ben-Gurion University of the Negev (BGU) and the Ph.D. degree in information systems from Tel-Aviv University. For the past 14 years, he has led the cooperation between BGU and Deutsche Telekom. He is currently the Director of the Telekom Innovation Laboratories, BGU, where he is also the Head of the Cyber Security Research Center and a Professor with the Department of Software and Information Systems Engineering. He is the Co-Founder of Morphisec, a startup company that develops innovative cyber-security mechanisms that relate to moving target defense. He has authored articles in leading peer-reviewed journals and in various peer-reviewed conferences. He has co-authored a book on social network security and a book on information leakage detection and prevention. His primary research interests are computer and network security, cyber security, Web intelligence, information warfare, social network analysis, and machine learning. He also consults professionally in the area of cyber security.