ThermoTag: A Hidden ID of 3D Printers for Fingerprinting and Watermarking

Yang Gao, Wei Wang, Yincheng Jin, Chi Zhou, Wenyao Xu, *Senior Member, IEEE*,
and Zhanpeng Jin, *Senior Member, IEEE*

***Abstract*— To address the increasing challenges of counterfeit detection and IP protection for 3D printing, we propose that every 3D printer holds unique fingerprinting features characterized by the thermodynamic properties of the extruder hot-end and can be used as a new way of 3D watermarking. We prove that these physical fingerprints resulting from manufacturing imperfections and system variations exhibit distinct heating responses, namely "ThermoTag," which can be represented as the distinguishable thermodynamic processes and, ultimately, the temperature readings during the preheating process. Experimental results show that, by only changing the hot-ends of the same model on the same 3D printer, we can achieve about 92% identification accuracy amongst 45 hot-ends. The permanence and robustness of ThermoTag for the same hot-end were examined, throughout a period of one month with hundreds of trials under different environmental temperature settings. Leveraging the hidden ThermoTag, an example of watermarking scheme in 3D printing is presented and evaluated.**

***Index Terms*— 3D printer, hot-end, thermal model, fingerprinting, watermarking.**

I. Introduction

ADDITIVE manufacturing (AM), also commonly known as 3D printing, is defined as a process of building up a three-dimensional object through forming successive layers of printing materials according to a digital 3D model. Technically, AM is capable of creating any shape with intricate internal structures and geometries. 3D printing has expanded and grown significantly in the past decade, due to the advancements in mechanical and materials technologies. According to the *Wohlers Report 2019*, the AM industry has a 21% CAGR (Corporate Annual Growth Rate), approaching $9.975 billion globally in 2018, and is expected to exceed $35.6 billion by 2024 [1].

With the significant advantages in short time-to-market, freedom to design, reduced tooling costs, and substantial material diversity, 3D printing has drawn increasing interest and attention from many industry sectors. It has also been applied in many safety- and mission-critical applications, including aerospace and defense, architecture and construction, biomedical fabrication, and automotive manufacturing [3]–[6]. SpaceX launched its Falcon 9 rocket with a 3D-printed part in its engine [7]. General Electric (GE) also planed to apply its first 3D-printed parts in its aircraft engine platform [8]. The U.S. Navy unveiled the first proof-of-concept, 3D-printed submersible hull, which was 90 percent cheaper and produced within a few days instead of the conventional 3-5 month [9]. On the other side, as consumer-grade 3D printers become more popular, affordable, and accessible, 3D printing has the potential to offer personalized products through mass customization, boost new designs and innovations through rapid prototyping, and support in-house manufacturing of consumer objects, parts, and components. Along with this incredibly transformative opportunity, the technology's biggest strength — the ability to easily create and distribute computer-aided design (CAD) files that convert the digital design model into a physical object — also poses a major challenge for design owners [10], [11]. 3D printing makes it easy to copy and reproduce products, because it is as simple as downloading a CAD file that can instruct the printer to reproduce a 3D object. Therefore, it is necessary to investigate new approaches to ensure the confidentiality and protection of intellectual properties (IPs) in 3D printing.

Among all the potential security risks associated with 3D printing, counterfeiting threats are becoming more prevalent and fundamentally different from threats usually causing manufacturing defects and product failures, as IP theft doesn't necessarily interfere with the 3D printing process or alter the mechanical and physical properties of 3D printed objects. Counterfeiting threats are mostly caused by the widespread illegal copying of the IP holder's design from either the cyber domain or the physical domain. Many researchers have been exploring the potential vulnerabilities and attack models for 3D printing [12]–[17]. But no matter for the malicious access to digital models or the side-channel reconstruction, the identification of IP is the most effective anti-counterfeiting method. Thus, to address the increasing threats related to IP theft in 3D printing, it is imperative to explore novel techniques for verifying the integrity of the 3D printing process and detecting the IP infringement.

Generally, for 3D printing, IP infringement can be considered as illegally duplicating 3D designs, copying the specific

Manuscript received July 7, 2020; revised January 6, 2021 and February 22, 2021; accepted February 28, 2021. Date of publication March 24, 2021; date of current version April 16, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marc Chaumont. *(Corresponding author: Yang Gao.)*

Yang Gao, Wei Wang, Yincheng Jin, Wenyao Xu, and Zhanpeng Jin are with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: ygao36@buffalo.edu; wwang49@buffalo.edu; yincheng@buffalo.edu; wenyaoxu@buffalo.edu; zjin@buffalo.edu).

Chi Zhou is with the Department of Industrial and Systems Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: chizhou@buffalo.edu).

Digital Object Identifier 10.1109/TIFS.2021.3065225

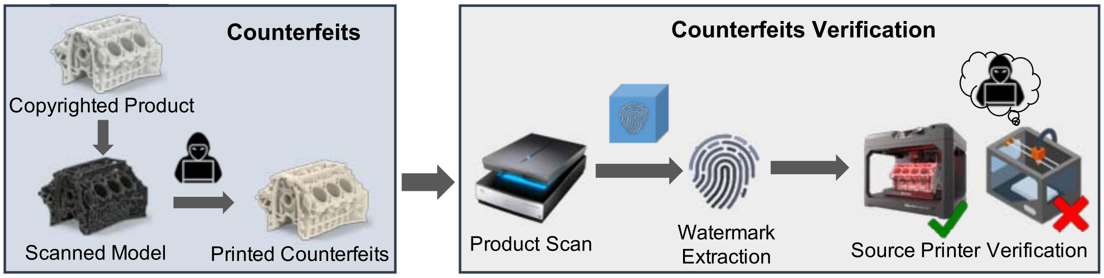


Fig. 1. The proposed counterfeiting threat model. (Left-side: malicious users illegally obtain the digital model of the copyrighted product with high-resolution scanners and fabricate counterfeits using unauthorized printers. Even with loss-less scanning, the re-printing process for different printers would still involve some uncontrollable variations in the watermark [2]. Right-side: the manufacturer and consumers could scan and extract the survived watermark, thus, to verify if the product is a genuine one or a counterfeit).

inherent features, or introducing external features. Some technologies such as 3D watermarking [18], [19] and adding anti-counterfeiting tags with Physically Unclonable Functions (PUFs) into the printed materials [20] have been proposed for IP authentication. PUFs are unique random physical patterns of taggants that cannot be copied and must be fabricated by a stochastic process, which however, come with the limitations of high computational and manufacturing complexity as well as the risk of changed material properties. Other solutions have also been explored by adding robust watermarks in 3D mesh models that can survive multiple attacks from the printing and scanning process [21], [22]. But for IP identification and protection purposes, those approaches need to inject a specific watermark or signature manually. Particularly for large volume production, storing and updating the digital watermark and signature database will increase the security risk and maintenance cost. Besides, if the attacker knows the decoding scheme or part of the watermarks, the watermarks on the products might be decrypted or manipulated, especially when manufacturers use the identical watermark for batch production. Just like the password in our daily life, it comes with high accuracy, but is also hard to remember and easy to forge, compared with the biometrics such as fingerprints.

Inspired by prior research that identified the fingerprinting features of various hardware devices [23], [24], we would like to ask the question: *Do 3D printers also possess their own fingerprints that cannot be erased and replicated?* Considering the heating system in a 3D printer, which controls the thermodynamic process in 3D printing that can directly influence the printing quality, we propose a hypothesis that, every heating system holds a unique and measurable feature that can make each 3D printer distinguishable, and thus can be used in a potential 3D printing counterfeit detection model (e.g., Figure 1). Specifically, our initial exploration of IP attacks on 3D printing focuses on the physical domain, in which unauthorized users can possibly scan the printed product and reconstruct it using their own 3D printers. In this article, we present a novel anti-counterfeiting method to extract the unique fingerprinting features of 3D printers, which can be used for counterfeit detection. Specifically, the unique, intrinsic fingerprinting features of each 3D printer will be seamlessly incorporated into the authentic 3D design model, to protect against unauthorized counterfeiting and piracy of products.

3D printing process can be generally divided into two phases: pre-heating and printing. A unique fingerprint of the legitimate 3D printer will be extracted during the pre-heating process and then inserted into the 3D design model using popular non-blinding watermarking methods. When unauthorized users seek to scan and reprint the 3D-printed, watermarked, genuine product, the hidden watermarks will be retained in the 3D-printed counterfeit products. Through the comparison of the hidden watermarks extracted from any 3D-printed product against the original user design model, the legitimacy of the printed product can be verified to prevent unauthorized IP infringement. Because the hidden watermark is generated based on the unique fingerprint of the genuine 3D printer, the attacker cannot obtain the same watermark by using any other 3D printers. In addition, each time the fingerprint generated from the same printer will be slightly different due to the environmental variance and thermal property of the hot-end itself. This dynamic trait of the fingerprint will help prevent reverse engineering. For instance, if the attacker decrypts one sample of the fingerprint by chance, he/she cannot apply the same fingerprint into other products or forge similar fingerprints without accessing the thermal model of the hot-end from the genuine 3D printer.

To the best of our knowledge, our work is the first of its kind to thoroughly investigate the thermodynamic process of 3D printing and discover its unique, measurable fingerprinting property. Leveraging this intrinsic device fingerprint in 3D printers, a more secure and robust anti-counterfeiting solution is presented to detect and protect against unauthorized IP infringement. Our contributions can be summarized as follows:

- We investigate the mechanical and thermal characteristics of hot-ends in 3D printers. The thermodynamics in the heating process unveils the uniqueness of each hot-end, which generates the 3D printer fingerprints. (Section IV)
- Fingerprinting features of 3D printers are designed and calibrated based on the rationale of ease of measurement and low computational cost. (Section V)
- Experimental evaluations are conducted based on three 3D printers of different models, 45 hot-ends of exactly the same model in the same printer, and hundreds of trials under different temperature settings. (Section VI)
- Through a real watermarking example, we demonstrate that the proposed technique is closely dependent upon the 3D printing process and the 3D printer itself, which

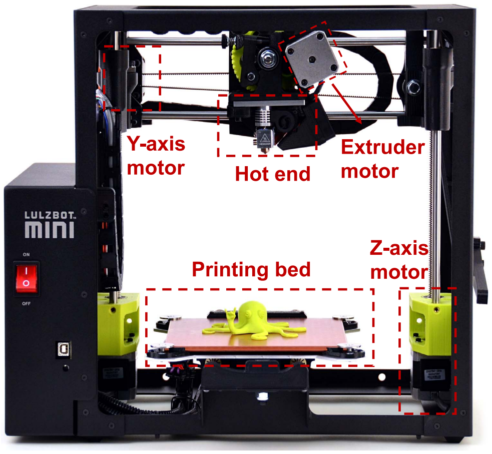


Fig. 2. A FDM 3D printer example (Lulzbot Mini), including a hot end, a extruder motor, Y and Z axis motors, and printing bed.

eliminates the needs of secure storage and encryption of PUF keys and watermark keys, and are more difficult to spoof and attack. (Section VII)

II. Background and Related Work

A. Overview of 3D Printing

Currently, there are many existing technologies applied in 3D printing, such as Fused Deposition Modeling (FDM), Selective Laser Sintering (SLS), Electron-Beam Melting (EBM), and Stereolithography (SLA). It was reported in the *Wohlers Report 2019* that [1] 591,079 consumer 3D printers were sold worldwide in 2018 (this may be an underestimation because it doesn't include those assembled from parts or those purchased as kits [25]), and most of them were based on FDM technology, more affordable and accessible compared with other methods. Meanwhile, thanks to the unrelenting efforts of the community on low-cost, open-source 3D printers [10], [26], such as the RepRap project [27] and the LulzBot 3D printers, it has seen a boom in entry-level 3D printing machines and the cost of 3D printers has decreased dramatically. Since 2010, the price of 3D printers that used to cost \$20,000 has dropped to a level of \$1,000 or less. According to a recent forecast report [28], the low-cost, sub-\$1000 desktop 3D printers will continue to be a major driving force for growth and is expected to grow at a rate of 12% into 2020. Therefore, in this article, we will focus on the FDM 3D printers.

FDM 3D printing follows exactly the principle of "additive fabrication" by laying down material in layers to produce a part, where a plastic filament or metal wire is unwound from a coil. Figure 2 presents an actual FDM 3D printer example (LulzBot Mini desktop 3D printer), which consists of X-axis, Y-axis, and Z-axis motors that control the movements of the extruder, a hot end for melting filament, an extruder motor controlling the filament's printing speed, and the printing bed right below the hot end for placing the printed object.

Figure 3 shows a complete 3D printing process consisting of five steps. First, users create a digital 3D model using Computer-Aided Design (CAD) software and convert it into a stand stereolithography (STL) file, which is widely used in rapid prototyping. Then, during the Computer-Aided Manufacturing (CAM) process, a layer description toolpath file (G-code) is generated by performing operations that includes slicing, path-planning, and support generation, by using 3D printing slicing tools such as Cura and Slic3r. Those toolpath instructions will be sent to the 3D printer to instruct the firmware to control the motor movement, the fan speed, and the extruder's printing speed to fabricate the desired product.

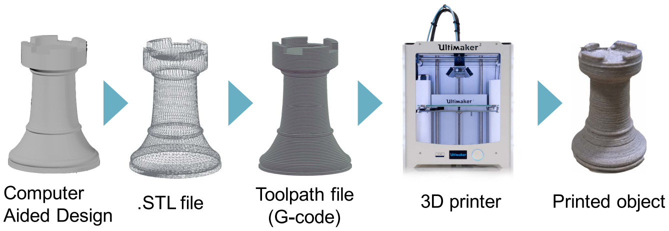


Fig. 3. 3D printing process chain.

B. Cyber Attacks on 3D Printing

With the fast growth of 3D printing market, cyber-security threats such as theft of 3D printing design or malicious manipulation of 3D printers have gained increasing attention [14], [29]. Sturm *et al.* [13] conducted a case study of multiple cyber-attack scenarios on the 3D STereoLithography (STL) files. Moreover, remote manipulation of 3D printers has been studied as another potential attack model [12]. In addition to the cyber domain, some research has been conducted to simulate and explore physical domain attacks on 3D printing. For example, researchers have presented thermal and acoustic side-channel attacks and reconstruction by using thermal cameras and audio recorders [30], [31]. Similar work has been done using acoustic waves and electromagnetic energy features that come from 3D printers as a side-channel attack [32], [33]. Compared with the side-channel attacks which have the limitation of physical proximity, scanning and printing are a more common and easier way to produce counterfeits. Especially for some products with lower printing resolution requirement, the counterfeited or replicated products can be easily obtained through reverse engineering. However, the corresponding defense scenarios are largely under-explored.

C. Hardware Fingerprinting of 3D Printers

Fingerprints were first used in the area of biometrics for identity verification. This concept subsequently appeared in the hardware security domain. Extrinsic fingerprints based on the inherent features of the device, so-called Physical Unclonable Function (PUF), were introduced to build the first silicon PUF authentication with integrated circuitry [34]. Since then, more and more silicon PUFs have been developed to verify the integrity of integrated circuits (ICs) and secure the critical components in hardware [35], [36].

With regard to the possible fingerprinting of 3D printing, a new method called *InfraStructs* [37] was proposed to insert common polymer material tags into 3D printed objects, which can be later read through Terahertz scanning (0.1 ∼ 10 THz). Instead of adding ID tags after 3D printing, a physical security feature for 3D printing was proposed by injecting quantum dots into PolyJet material before printing [38]. According to Brownian motion, these nanoparticles were randomly distributed. By observing the quantum dots' light spectrum distribution after absorbing UV light, this PUF allowed every 3D-printed object to be identifiable and distinguishable. A recent study also shows the feasibility of utilizing plasmonic nanopaper as PUF tags for anti-counterfeiting applications [39].

In addition, the manufacturing imperfection and associated system operational variations also could be used as the fingerprints of 3D printers [2]. The distortion of material extrusion caused by the stepper motor could be reflected via the uniqueness of the printed QR code that can be captured through a commercial smartphone [40]. Prior studies have also investigated the uniqueness of inherent equipment distortions to track the source printer of a 3D printed object [41].

D. 3D Watermarking

Another possible fingerprinting method for 3D printers is 3D watermarking [42]. A 3D printer with a fine stair-step can ensure that the printed object's distortions, such as surface texture and roughness, are reduced to a microscopic level. Using the high-resolution Kinect or other 3D scanners, it is possible to reconstruct a 3D model through scanning. There are two major types of watermarks. The first type is visible watermarks (e.g., layer artifacts, printing deformations). It is easy to identify, but meanwhile, it is not safe for the copyright holder when malicious attackers attempt to erase the watermark. The second type is invisible watermarks (e.g., mesh-based). The signature may be lost or removed during the common model processing steps, such as lossy compression and simplification, which is usually applied to the 3D object. Zafeiriou *et al.* [43] proposed two robust blind watermarking schemes. The first method, called Principal Object Axis (POA), embedded the signature by modifying a set of vertices that correspond to specific angles θ. And the second method, called Sectional Principal Object Axis (SPOA), displaced the set of vertices having the coordinate θ domain within a specific ranges. Cho *et al.* [44] proposed a robust blind watermarking method by shifting the mean or variance of the histogram distribution of the mesh vertex norms. This method is fairly robust against various distortion attacks; however, it would also cause visible artifacts on the surfaces of 3D models. Based on Cho's algorithm, Yang *et al.* [45] developed a modified method by changing the discrete statistic feature which is the height of the histogram bins, instead of the mean and variance. Hou *et al.* [46] proposed a robust and blind watermarking scheme in 3D-printed models by utilizing the layering artifacts. Its printing-axis estimator for alignment needs strong printing artifacts, whereas the watermark extraction process may be fragile to those strong printing artifacts.

III. Theoretical Model

We consider an adversary that aims to compromise the IP and even possibly claim the copyright of a 3D printed design by illegally scanning and replicating the IP owner's 3D-printed products. It is reasonably assumed that the adversary has neither any knowledge about the original 3D design model nor any access to the legitimate 3D printer. This is a very common scenario that happens not only in the additive manufacturing domain but also in the entire traditional manufacturing industries. Although there have been extensive research efforts and a wide range of technologies to ensure the security of digital design files and reduce the risk of attacks on the critical manufacturing infrastructure, it is unpreventable and inevitable that the adversaries can physically scan the 3D-printed products to acquire the full geometry and details of a 3D design and then replicate the design using their own 3D printers. The recent advancements in high-resolution, high-precision 3D scanning systems make the IP protection more challenging and more critical than ever.

A. Methodological Flow

The proposed 3D printer fingerprinting process contains three phases: 1) pre-heating the hot-end to the target temperature; 2) extracting the fingerprinting features from the temperature reading; 3) send the extracted fingerprinting features back to the slicing software to be embedded into the 3D design model as the watermark.

1) Phase 1: Pre-Heating (See Section IV): Pre-heating process is used for melting filament, such as PLA and ABS, which is the prerequisite for FDM 3D printing. Our proposed fingerprint is generated from a normal printing process without adding any extra hardware.

2) Phase 2: Fingerprint Extraction (See Section V): The temperature readings during the pre-heating process (usually around 1 minute) will be recorded, and the corresponding fingerprinting features will be extracted using a sparse autoencoder.

3) Phase 3: Watermark Embedding (See Section VI): To ensure that the fingerprints will survive through the slicing process, even for some low-resolution printing setting, the design file will be embedded with the fingerprint as watermarks after being sliced as the toolpath file (G-code).

Assumptions: Because the hot-ends are not consumables and well-maintained in a carefully packaged manner, it is arguably assumed that the hot-ends in the 3D printer will not be frequently replaced during the operational lifetime.

B. Definitions

1) Definition 1: Intellectual Property (IP): For a 3D printing process, let C denote the whole IP information for the design model. c_i represents each independent attribute that determines the quality of a 3D printed model, such as the vertex placement, the mesh property, the infill structure, the layer thickness, and the printing temperature.

$$C = \{c_1, c_2, \ldots, c_k\} \tag{1}$$

2) Definition 2: Physical Observations: Let O represent the variable set of all physical information that can be sensed and measured from the 3D printer during the entire printing process. Each o_i is considered as an individual physical observation, including the nozzle's temperature, the motor's vibration, or other side-channel information.

$$O = \{o_1, o_2, \ldots, o_m\} \tag{2}$$

3) Definition 3: Fingerprint Extraction: Let P be the set containing each individual fingerprinting feature p_i that resulted from manufacturing imperfection or distinct behaviors of different parts in 3D printers. And $g(.)$ is the fingerprint extraction function and is determined through the analysis of mapping relation between each fingerprinting feature and the corresponding observation.

$$P = \{p_1 \leftarrow g_1(o_1), p_2 \leftarrow g_2(o_2), \ldots, p_m \leftarrow g_m(o_m)\} \tag{3}$$

4) Definition 4: Watermark Embedding and Detection Function: Let $\mathcal{F}$ and $\mathcal{F}'$ indicate the watermark embedding and detection schemes respectively. And $\hat{C}$ denotes the IP information of the watermarked 3D model.

$$\hat{C} = \mathcal{F}(C, P) \tag{4}$$

$$P = \begin{cases} \mathcal{F}'(\hat{C}) \Rightarrow \text{blind watermarking} \\ \mathcal{F}'(\hat{C}, C) \Rightarrow \text{non-blind watermarking} \end{cases} \tag{5}$$

5) Definition 5: Printing and Scanning Process: For watermarking, the printing and scanning process can be both considered as the attacking function that might affect the watermark stored in the target model. Let $\mathcal{H}$ and $\mathcal{H}'$ represent the printing process and the scanning process, respectively. During the printing and scanning process of the watermark P, considering the printing noise ϵ_p and observation noise ϵ_r during the scanning process, we define

$$\mathcal{H} \rightarrow P + \epsilon_p \tag{6}$$

$$\mathcal{H}' \rightarrow P + \epsilon_r \tag{7}$$

where the printing noise ϵ_p caused by the material's thermal property and the step motor's vibration was proven to be quite subtle and random according to the literature [2], [47]. Hence, we neglect this noise in this work. The noise ϵ_r introduced by image recording and processing usually belongs to the Gaussian noise with a normal distribution of zero mean. Thus it can be roughly removed by processing multiple observed images for the same watermark.

C. Problem Formulation

1) Formulation 1: Model Watermarking: The purpose of model watermarking is to protect the copyright of the 3D model and generate the secure watermarks based on the unique intrinsic characteristics of the 3D printer itself. C_w is defined as the IP of the printed watermarked model.

$$C_w = \mathcal{H}(\hat{C} \leftarrow \mathcal{F}(C, P)) \tag{8}$$

2) Formulation 2: Model Verification: We assume that the attacker can obtain a 3D printed product that holds the IP C_w and manufacture counterfeits through scanning and reprinting the acquired 3D design model using any available 3D printers. To verify the ownership and legitimacy of the IP C_u of a product that is printed according to the scanned 3D design model ($\mathcal{H}'(C_w)$) acquired by the attacker, we define the IP assessment based on the pre-defined watermark P. When the pre-defined watermark P is decoded, we can consider it as authentic one, otherwise, it is a counterfeit.

$$C_u = \mathcal{H}(\mathcal{H}'(C_w)) \tag{9}$$

$$\mathcal{F}'(C_u) = \begin{cases} = P \Rightarrow \text{Authentic} \\ \neq P \Rightarrow \text{Counterfeit} \end{cases} \tag{10}$$

IV. Thermodynamics of Hot-Ends in 3D Printing

This section provides a brief introduction to the hot-end in 3D printers in terms of its physical structure and mechanical properties. Followed by this, we present preliminary experimental results to support our hypothesis about hot-end-based fingerprinting of 3D printers. A detailed analysis will be presented in the later sections.

A. Thermal Modeling and Variations of 3D Printer Hot-Ends

Hot-end is part of the extruder in 3D printers and is in charge of melting filament such as Polylactic Acid (PLA) or Acrylonitrile Butadiene Styrene (ABS). A typical hot-end consists of four pieces, including the cartridge heater, thermistor, metal frame, and nozzle. Figure 4 shows the physical structure and the thermal conduction process of an hot-end in the Prusa i3 HIC 3D printer. A general three-dimensional thermal conduction process in an isotropic can be modeled using the following equation:

$$\frac{\partial u}{\partial t} = \frac{k}{C_p \rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + q(t) - hA(u - u_e) \tag{11}$$

$$q(t) = C_h \mathrm{PWM}(t - d) \tag{12}$$

where in Equation 11, u is the temperature as a function of space and time; $\frac{\partial u}{\partial t}$ denotes the rate of change in temperature at a certain time point; $q(t)$ is the heat generated from the cartridge heater, and then conducted to the thermistor at the location $\{x, y, z\}$ which eventually result in the temperature change in the nozzle; k is the thermal conductivity; ρ is the material density; C_p is the specific heat capacity; $\frac{k}{C_p \rho}$ is called thermal diffusivity; h is the heat transfer coefficient by heat convection with air; A is the area of the object exposed to the air; and u_e is the environmental temperature. In Equation 12, C_h is the heat capacity of the cartridge heater; PWM is the Pulse-Width Modulation, which refers to the control variable of the heater for heat generation, and d is the time of heat conduction inside the heater. Based on this simplified model, it is shown that many intricate configuration factors may lead to a unique, complex thermal conduction process of an individual hot-end. Different hot-ends have diverse dimensions and use various materials for the metal block. Even with the same material, the thermal diffusivity may vary due to the manufacturing imperfections [48]. In addition, resistors' tolerance in the heater will cause a self-heating effect that further influences the temperature change rate.

B. Evidence of Hot-End Uniqueness

To preliminarily prove the feasibility of using hot-ends for hardware fingerprinting purpose, we set up an experiment using three desktop 3D printers of different models: 1) Lulzbot

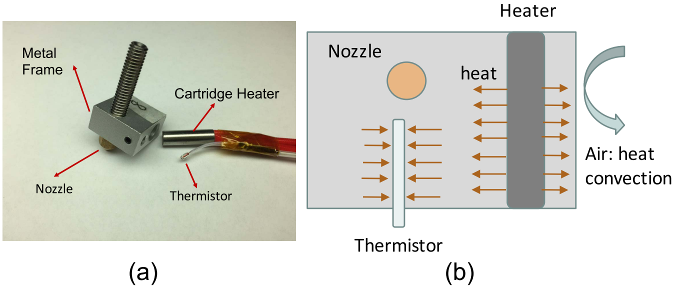


Fig. 4. (a) RepRap Prusa i3's hot end includes metal block, a MK8 0.4mm nozzle, a 12V 40W cartridge heater and a 100K NTC thermistor; (b) Simplified thermal conduction model.

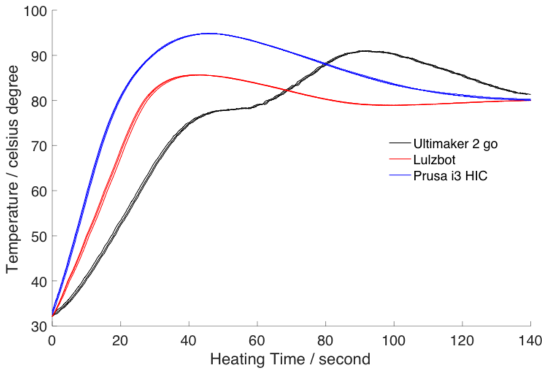


Fig. 5. Temperature curves from three different printers during pre-heating.

Mini, 2) Prusa i3 HICTOP, and 3) Ultimaker 2 Go, with the same default Proportional-Integral-Derivative (PID) setting for temperature control. Then we pre-heated the hot-ends to 80 degrees Celsius (°C) and obtained the temperature readings at a sampling rate of 50 Hz during the heating process by accessing the Arduino board in 3D printers. We repeated this process five times to reduce random variations. Figure 5 depicts the preheating temperature curves for three different types of printers (represented in three colors), and five temperature curves for each printer (highly overlapped cluster). It is clearly shown that different models of printers have observable and distinguishable behaviors and characteristics in the temperature curves during the pre-heating process.

To validate our hypothesis about the uniqueness of 3D printers, particularly the critical hot-end, we would like to ask a question: *Can the 3D printers of the same model hold unique hot-end-based fingerprints?* To answer this question, we investigate the temperature control system in the 3D printer as shown in Fig. 6. Based on the nozzle's current temperature at time t, Temp_k, the Pulse-Width Modulation (PWM) which is the control variable of the heater is calculated as below:

$$\text{error}_t = \text{Temp}_{\text{target}} - \text{Temp}_t \tag{13}$$

$$\text{PWM}_t = K_p \text{error}_t + K_i \int_0^t \text{error}_t dt + D\text{term}_t \tag{14}$$

$$D\text{term}_t = (1 - K1) * K_d \frac{d\text{error}_t}{dt} + K1 * D\text{term}_{t-1} \tag{15}$$

where the PWM_k depends on the temperature error_k through three individual PID elements. 1) The *(P)roportional term* is based on the default parameter K_p and the instant error. The higher the P term is, the faster the output approaches the target value, while a rather high K_p value may result in overshoots. 2) The *(I)ntegral term* is based on the default parameter K_i and the accumulative error in a certain period of time. A suitable K_i can help reduce steady-state error. 3) The *(D)eviation term* is responsible for predicting the future based on the current error change rate, and a higher K_d will increase the system responding speed. Factor $K1$ can make the Dterm change smoothly and reduce the temperature's overshoots. In our 3D printer temperature control system, $K1$ is set as 0.95.

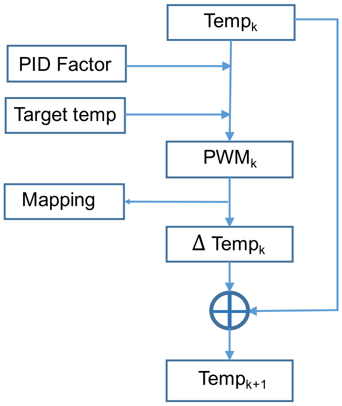


Fig. 6. Temperature control flow.

As discussed in Section IV-A, the heat conduction process will bring in many manufacturing imperfections, which result in the variations of thermodynamic characteristics in hot-ends. In addition, the heat generation $q(t)$ is also not identical among different heaters of the same type. Therefore, our hypothesis is that, each individual hot-end will hold a unique mapping function from the PWM variables to the temperature changes,

The temperature only depends on three attributes: 1) default PID setting, 2) target temperature, and 3) the hot-end's thermodynamic mapping function. Among these three attributes, the PID setting and target temperature are usually device-dependent and shall be determined according to the specific 3D printing task. The thermodynamic mapping function of the hot-end, which results from the individual hot-end's manufacturing imperfections and system variations, is believed to be the key component that influences the heating process during the 3D printing. Such mapping functions shall be examined from two perspectives: uniqueness and consistency. In terms of uniqueness, the mapping function should be distinguishable among different hot-ends, even with exactly the same default PID setting and target temperature. Consistency means that the mapping functions generated from the same hot-end should be consistent with different default PID settings or target temperatures.

The thermistor's location within the hot-end is fixed and the second derivation of temperature u by the x, y, z location is a constant. The typical h ranges from 5 - 10 $W/(m^2K)$, and the surface area of the hot-end is usually limited within 16 cm^2. Thus, the power of the entire heat convection approximately ranges from 0.48 W to 0.8 W, which can be neglectable compared with the 40 W heating power. Therefore, the heat

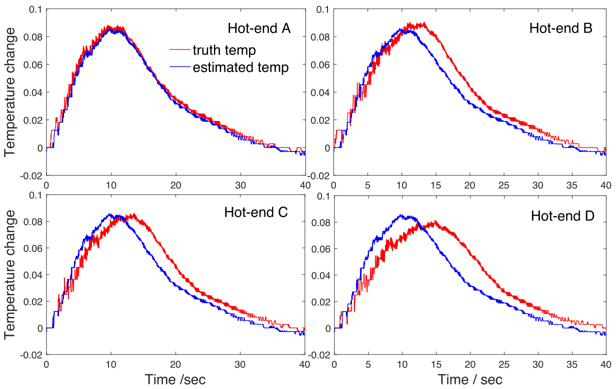


Fig. 7. Ground truth and estimated temperature changes of 4 hot-ends for the same setting (A is the same hot-end for the training, and B, C, and D are three different hot-ends).

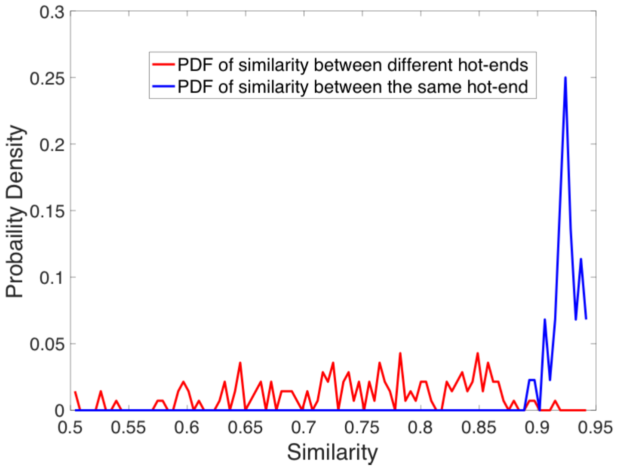


Fig. 8. Similarity distribution for the same hot-end and 29 different hot-ends.

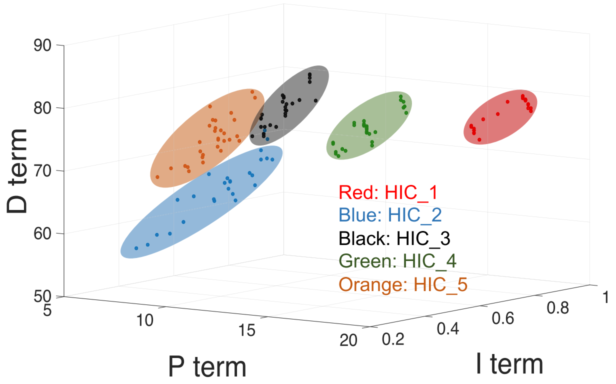


Fig. 9. Well-tuned PID values for different hot-ends clustered in 3D space with clear separation.

conduction can be further simplified as follows:

$$\frac{\partial u}{\partial t} = \Delta\text{temp} = \alpha + \beta\text{PWM}(t-d) \tag{16}$$

$$\alpha \propto \frac{k}{C_p\rho}, \beta = C_h \tag{17}$$

where the specific heat capacity C_p and C_h vary over different temperatures. Therefore, the relationship between Δtemp and PWM$_t$ is a non-linear mapping function.

To better model the non-linear thermodynamic mapping function of the hot-end, we adopt a simple Multi-Layer Perceptron (MLP) neural network with two hidden layers and 50 neurons per layer. In the training phase, we let the MLP learn to approximate the relationship between the input set, which is one hot-end's PWMs during a certain period of time, and the output set which is the same hot-end's temperature changes. In the testing phase, we can obtain the estimated temperature changing curves based on the testing PWM readings. The correlation coefficient is used to describe the similarity between the estimated and ground-truth temperature change curves, which is defined as:

$$\rho(A,B) = \frac{cov(A,B)}{\sigma_A\sigma_B} \tag{18}$$

where A and B represent the estimated and ground-truth temperature curve respectively, $cov(A,B)$ is the covariance of two temperature curves, σ_A and σ_B are the standard deviations of the estimated temperatures and the ground truth temperatures respectively.

1) Be Unique Among Different Hot-Ends: In the experiment, we set the default PID parameters as the fixed values, where $K_p = 11.95$, $K_i = 0.49$, $K_d = 72.55$, and the target temperature as 80 degree Celsius (°C). By modifying the firmware, we collected the temperature and PWM readings from 30 different hot-ends of exactly the same model, and conducted five times of heating process for each hot-end. One of the hot-ends was randomly selected as the genuine one, for which we further repeated the heating process and collected temperature and PWM readings for 60 more times. The MLP was trained based on the genuine hot-end's PWM and temperature readings. In the testing, the estimated temperature curves for the other different 29 hot-ends were generated through the employed MLP based on the corresponding hot-end's PWM readings as inputs, shown in Figure 7. For the same hot-end A, the estimated temperature change curve fits the ground-truth temperature curve very well and shows a weak similarity with another three different hot-ends, B, C, and D. Those differences largely result from the difference of the manufacturing imperfections among those hot-ends. The more considerable difference of physical characteristics (discussed in Section IV.A) between the target hot-end A and test hot-end (B, C, or D), the more clear difference shown in Fig. 7.

Based on the 145 similarity values between the 29 different hot-ends and the single, randomly chosen genuine hot-end, as well as the 65 similarity values between any two arbitrary trials of the same genuine hot-end, we plotted the Probability Density Function (PDF) in Figure 8. It is manifest that, the similarity between any two trials of the same hot-end is largely concentrated on a rather high level, i.e., 0.9 ∼ 0.95, while the similarity levels between the chosen hot-end and the other 29 hot-ends are widely spread out throughout the range of 0.5 to 0.9. The clearly distinguishable difference of similarity distribution indicates that the thermodynamic mapping function from the hot-end holds a high level of uniqueness and robustness.

TABLE I

TUNING PARAMETERS FOR ZIEGLER-NICHOLS METHOD [49]

Control type	K_p	K_i	K_d
P	$K_u/2$	-	-
PI	$K_u/2.2$	$1.2K_p/T_u$	-
classic PID	$0.60K_u$	$2K_p/T_u$	$K_pT_u/8$
Pessen Integral Rule	$0.70K_u$	$2.5K_p/T_u$	$0.15K_pT_u$
Some overshoot	$0.33K_u$	$2K_p/T_u$	$K_pT_u/3$
No overshoot	$0.33K_u$	$2K_p/T_u$	$K_pT_u/3$

2) Be Consistent for the Same Hot-End: In order to evaluate the consistency of the thermodynamic mapping function from the same hot-end, based on Figure 6, we conducted a series of experiments by using different default PID parameters and different target temperatures. Because the PID parameters have a significant impact on the temperature oscillation, and as shown in Figure 9, different hot-ends have clearly separable PID clusters. In order to mimic the real-world scenarios, a default PID parameters set $PID = \{pid_1, pid_2, \ldots, pid_k\}$ for the genuine hot-end is generated by an optimized auto-tuning algorithm available in the Marlin firmware.

For 3D printing, the temperature control system has a relatively low requirement on the decay ratio during the steady-state oscillation and is not sensitive to overshooting. Thus most 3D printer controllers adopt the Ziegler-Nichols tuning method [49]. This method (shown in Table I "Control type: classic PID") disables I and D gains initially, and only increases K_p to the ultimate gain K_u, at which the system keeps doing undamped oscillations. Finally, the P, I, and D gains are determined by the ultimate gain K_u and the oscillation period T_u. K_u denotes the ultimate gain for loop stability, and T_u is its corresponding period. We also list other PID tuning settings under different control preferences in Table. I. To gain more stabilized, auto-tuned PID values, based on the tuning setting "Control type: classic PID", we optimized the existing PID tuning algorithm (as shown in Algorithm 1) available in the Marlin firmware, an open-source firmware for many desktop 3D printers.

We performed the experiments under three different target temperatures of 80, 90, and 100 degrees Celsius respectively. For each target temperature setting, we collected temperature and PWM readings for 12 different PID parameters and repeated the experiments by five times. Thus, in total, we have 180 pairs of data for the target temperature settings and PWM readings from the same hot-end.

C. Temperature-PID Mapping Function

The Proportional-integral-derivative (PID) controller is a classic and widely used temperature control mechanism based on the feedback theory [50]. A typical equation for PID control is as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{19}$$

where the $u(t)$ denotes the control variable which depends on current error $e(t)$ through three individual elements. 1) The *proportional term* K_p takes charge of the instant error. The higher the p term, the faster the output approaches the target value, while a rather high K_p value may result in overshoots. 2) The *integral term* K_i is based on the accumulative error in a certain period of time. A suitable K_i term can help reduce steady-state error. 3) The *derivative term* K_d is responsible for predicting the future based on the current error change rate, and it will increase the system responding speed and reduce the undesired overshoots.

Algorithm 1: Optimized Auto-Tuning Algorithm

```
Input: $S_t$: target temperature;
  $I_t$: target iteration times; $I_c$: current iteration times;
  $S_c$: current temperature;
  $T$: minimum oscillation period;
Output: $P_i, I_i, D_i$: Optimized PID values for $I_t$
        iterations
Start heating at the default PWM value R;
if $S_c > S_t$ and time $> T$ then
    Stop heating;
    Record period for upper peak $T_{up}$ and maximum
     temperature $S_{max}$;
end
if $S_c < S_t$ and time $< T$ then
    Start heating;
    Record period for lower peak $T_{down}$ and minimum
     temperature $S_{min}$;
    R $= \frac{T_{up}-T_{down}}{T_{up}+T_{down}}R + R$ ; // update R values;
    if $I_c < I_t$ then
        $K_u = \frac{8R}{\pi(S_{max}-S_{min})}$;
        $T_u = \frac{T_{up}+T_{down}}{samplingrate}$;
        $I_c++$;
        Calculate $P_i, I_i, D_i$;
    end
    Autotune finish;
    Return $P_i, I_i, D_i$;
end
```

D. Estimated Fingerprint Capacity

Given a digitized fingerprint authentication system for M users, its constrained capacity [51] can be denoted as C:

$$C = \frac{1}{2}\log_2[1 + \frac{\overline{d_m^2}}{4\max(\overline{\sigma_g^2}, \overline{\sigma_i^2})}] \tag{20}$$

where $\overline{d_m}$ measures the distance between the median matching scores from the genuine PDF distribution $\hat{g}_m$ and the imposter PDF distribution $\hat{l}_m$. Both $\hat{g}_m$ and $\hat{l}_m$ belong to the normal distribution $N(0, \overline{\sigma_g^2})$ and $N(0, \overline{\sigma_i^2})$, respectively.

Given the matching score from 0 to 1 for 45 different hot-ends, our result shows that the constrained capacity of our fingerprint system C is 0.313 with $\overline{d_m} = 0.17$, $\overline{\sigma_g^2} = 0.0072$, $\overline{\sigma_i^2} = 0.0133$, which indicate that our system is reasonably accurate and scalable, compared with $C = 0.33$ for the palmprint and $C = 0.22$ for the hand geometry reported in [51].

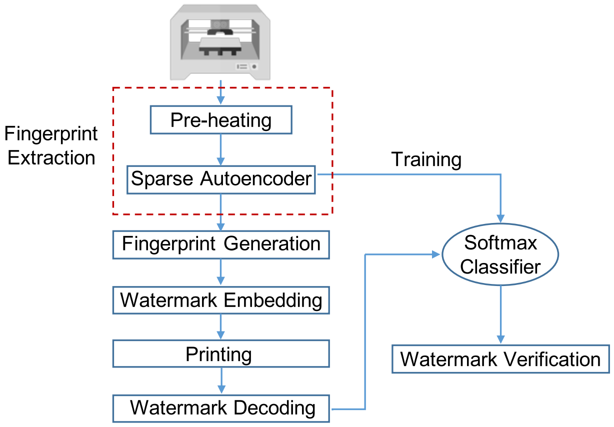


Fig. 10. 3D printer fingerprinting based on preheating temperature readings and auto-tuned PID values.

V. ThermoTag-Based Fingerprinting

This section describes how the fingerprinting features of 3D printers are extracted and verified based on the uniqueness of hot-ends, including: 1) fingerprint selection, 2) data collection, 3) fingerprint generation, and 4) fingerprint matching. The methodological flow diagram is shown in Figure 10. During the enrollment phase, when the 3D printer is performing the printing task, the pre-heating temperature reading is recorded to train the verification classifier. After finishing the enrollment, when the 3D printer receives the new printing task, the hot-end starts to pre-heat to the target temperature. The corresponding temperature reading is captured and extracted into the fingerprint via the autoencoder. The fingerprint is further embedded as the watermark in the object during the printing process. To verify the identity (i.e., whether it is printed by the authorized source) of an unknown printed product, we scan and decode the watermark in the product. The extracted watermark is inputted into a softmax classifier to verify the legitimate printing source.

A. Fingerprint Selection

In Section IV-B, we have demonstrated the existence of the fingerprints of hot-ends by defining the uniqueness of thermodynamic mapping function from the heater's PWM to the hot-end's temperature change. However, as an effective hardware fingerprint, it needs to be ease of implementation and low computational cost. As the temperature change can be defined in the following way:

$$\Delta\text{temp} = g(f(\text{PID}, \text{temp}_T)) \tag{21}$$

where Δtemp is the temperature change, g is the mapping function from PWM to the hot-end's temperature change. f is the temperature tuning function. PID is the default PID parameters, and temp_T is defined as the target temperature.

In the real scenario, to ensure a good and stable printing quality, each individual 3D printer will hold a fixed, well-tuned PID parameter set and a fixed target temperature for the pre-heating process. Correspondingly, in Equation 21, with the same temperature tuning function f, PID and temp_T, Δtemp only depends on the mapping function g.

Hence, we propose the "ThermoTag" fingerprints, which make use of the uniqueness of the temperature change curve during the pre-heating process to represent the uniqueness of thermodynamic mapping function resulted from the manufacturing imperfections and system variations of hot-ends.

B. Data Collection

By accessing the MKS V1.4 controller board in the 3D printer, we obtained the temperature readings of the hot-end by the built-in 100 Kohm NTC thermistor during the pre-heating process, with 10, 15, 25 and 50 Hz sampling rates respectively.

An intuitive question is, *how long should the temperature reading be recorded to capture the uniqueness?* The entire pre-heating process is composed of heating, cooling, and oscillating stably to the target temperature. As the uniqueness we defined is caused by the heat generation and conduction behaviors, and the cooling behavior is more affected by the room temperature; thus, we will focus on the very first 50 second heating period.

C. Fingerprint Generation

1) Temperature Pre-Processing: Instead of directly extracting the fingerprint features from the raw temperature data which often starts from sightly different room temperatures (20~25 degree Celsius), we normalized the temperature curves by filtering out the temperature below 30 degree Celsius.

2) Feature Extraction: We design our ThermoTag by using the sparse auto-encoder to automatically extract a comprehensive set of intrinsic features from the temperature curves through an unsupervised learning process.

A regular autoencoder is a neural network which is trained to reconstruct the input and compress the input data by using the limited neurons in the hidden layer [52]. In a standard autoencoder with one hidden layer, we can assume that the input $x \in R^p = \chi$ is mapped to the hidden layer $z \in R^q = \zeta$ as follows:

$$z = h_{hidden}(Wx + b) \tag{22}$$

z is subsequently mapped onto output layer x':

$$x' = h_{out}(W'z + b') \tag{23}$$

where h denotes the activation function of neurons.

The training process is based on the optimization of the cost function $J(W, b)$, which minimizes the reconstruction error. In our method, given the training set $S = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ representing the N different hot-end temperature curves, the weight matrix W and bias vector b are used to generate a new labeled training set $\hat{S} = \{(\hat{h}(x^{(1)}), l^{(1)}), \ldots, (\hat{h}(x^{(N)}), l^{(N)})\}$, where $\hat{h}(x^n)$ is the activation vector obtained in the hidden layer, and l^n represents the corresponding labels.

The sparse autoencoder is a specialized autoencoder that adds a sparsity regularizer to its cost function (mean squared error) [53].

$$J_{sparse}(W, b) = J(W, b) + \beta * \Omega_{sparsity} \tag{24}$$

where $J(W,b)$ is the original cost function, e.g., quadratic cost, cross-entropy, or exponential cost, and β is the coefficient for the sparsity regularization.

The sparsity regularization is to minimize and constrain the activation values by comparing the Kullback-Leibler divergence.

$$\begin{aligned}\Omega_{sparsity} &= \sum_{j=1}^{M} KL(\rho||\hat{\rho_j}) \\ &= \sum_{j=1}^{M} \rho log(\frac{\rho}{\hat{\rho_j}}) + (1-\rho) log(\frac{1-\rho}{1-\hat{\rho_j}}) \qquad (25)\end{aligned}$$

where M is the number of neurons in the hidden layer and ρ is the desired activation value.

The sparsity regularizer based on the average activation value of jth neuron in the hidden layer is defined as:

$$\hat{\rho_j} = \frac{1}{N}\sum_{i=1}^{N} h(\omega_j^{(1)T} x_i + b_j^{(1)}) \qquad (26)$$

where N is the total number of training examples.

With a low output activation value, this jth neuron is considered to respond to a few training examples, which means that in the hidden layer, each neuron is associated with some features representing a specified subset of total training samples.

D. Fingerprint Matching

In order to distinguish the pre-heating behaviors among different hot-end configurations, an unsupervised feature learning and classification method is proposed by combining the sparse autoencoder and the softmax classifier layer.

We further classify those labeled features $\hat{S} = \{(\hat{h}(x^{(1)}), l^{(1)}), \ldots, (\hat{h}(x^{(N)}), l^{(N)})\}$ (extracted from the temperature curves) by training a softmax classifier, which has been proved to be effective for multi-class classification problem. A softmax function is given as:

$$P(y=j|x^{(i)},\theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{k=1}^{K} e^{\theta_k^T x^{(i)}}} \qquad (27)$$

where j is the class type, and θ is the model parameter to minimize the cost function by using gradient descent. The cost function is shown as below:

$$J(\theta) = -\left[\sum_{i=1}^{N}\sum_{j=1}^{K} 1\{y^{(i)}=j\} log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{k=1}^{K} e^{\theta_k^T x^{(i)}}}\right] \qquad (28)$$

where K is the number of different classes.

When the feature vector of a new temperature reading curve is presented to the well-trained softmax classifier, its most likely output will be the class (i.e., the specific 3D printer unit) which has the maximum probability $\arg\max_{y} p(y=j|x^{(i)},\theta)$.

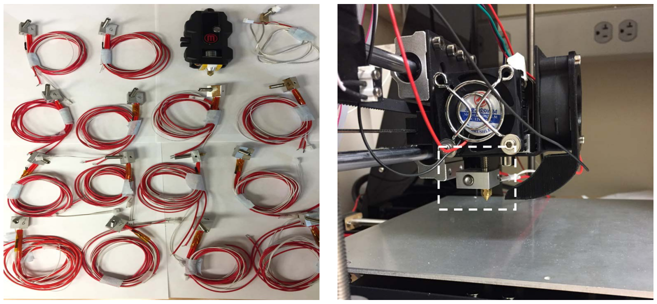

(a) Experimental hot-ends of the same model and different models (b) Hot-end setup in the Prusa i3 3D printer

Fig. 11. Experimental device and measurement setup.

VI. Performance Evaluation

A. Experimental Setup

1) Standalone Hot-End Setup: In the previous section, a small scale experiment among 3 different types of 3D printers has been conducted. The result shows distinguishable differences among those printers. Here we will focus on a more challenging task: to evaluate the fingerprinting features extracted from the hot-ends of exactly the same model.

To prove that 3D printers can be distinguished only by the equipped hot-ends, we conducted the experiments using 45 individual hot-ends on a single Prusa i3 HICTOP 3D printer. The experimental setup is shown in Figure 11. To further evaluate and verify the uniqueness of hot-ends resulted from the manufacturing imperfections, the chosen 45 hot-ends of exactly the same model were acquired from the same vendor and the same manufacturing assembly line.

2) Temperature Setup: As we have demonstrated in Section IV-B, the target temperature has no influence on the uniqueness of the intrinsic fingerprintings in the hot-ends. To reduce the cooling period between each test less than 20 minutes, we set 80 degree Celsius as our target pre-heating temperature.

3) Data Collection and Allocation Setup: The data collection protocol is described in Section V-B. As the variances of the temperature curves for the same hot-end in different trials are very small and trivial, thus for each hot-end, we collected the whole temperature readings of the 3D printer, from the initial room temperature to the final stabilized target temperature by only five times. Totally, we have 225 trials for the 45 individual hot-ends.

In the evaluation phase, we evaluated our system for five times. Each time for each hot-end, we randomly selected three trials as the training set and the remaining two trials as the testing set. Thus, in total, we have 135 trials for the training and 90 trials for the testing.

B. Metrics

To evaluate a hardware fingerprint, we want first to know how each fingerprint can be distinguishable from each other. Let k be the total number of hot-ends. After training the softmax classifier with all the k hot-ends, given a testing preheating temperature trial, we will have an estimated label for

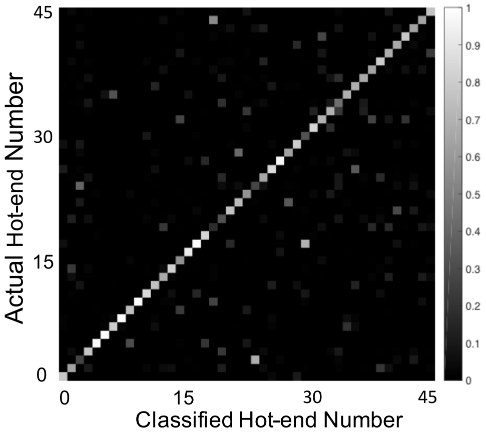


Fig. 12. Confusion matrix over 45 hot-ends.

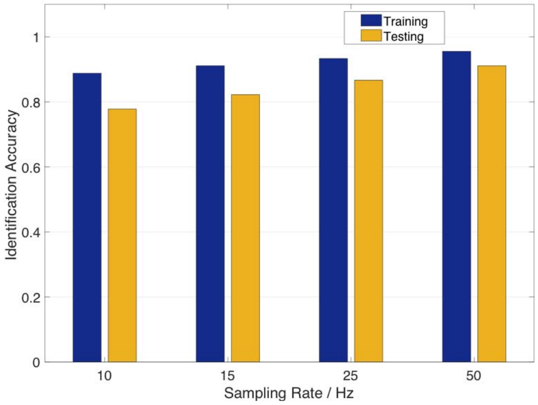


Fig. 13. Identification accuracy among 45 hot-ends at different sampling rates.

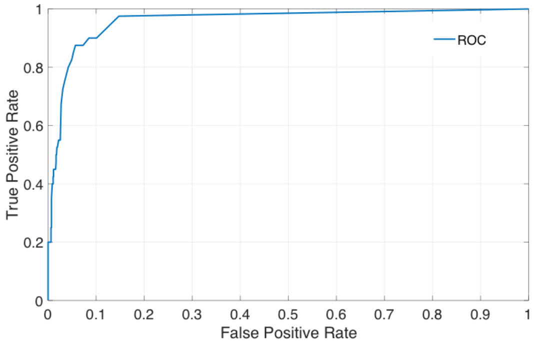


Fig. 14. Average ROC curve for trained 20 classes.

one of k hot-ends. Based on the ground truth label, we define the identification accuracy as below:

$$\text{Accuracy} = \frac{\sum_{i=1}^{k} CT_i}{N} \tag{29}$$

where CT_i is the number of correctly identified trials for the ith hot-end, and N is the total number of testing trials.

In addition to the multi-class classification performance, we also investigate the performance of our fingerprinting scheme with the existence of alien hot-ends (i.e., unknown devices for the 3D printer system). We define FP as the number of false positives, which means the number of incorrectly accepted trials. m and n are the number of trained classes and alien classes. An error evaluation metric which equals to the average false acceptance rate (FAR) are defined as below:

$$\text{Error (Avg. FAR)} = \frac{\sum_{i=1}^{m} FP_i + \sum_{j=1}^{n} FP_j}{N} \tag{30}$$

C. Performance

Firstly, we trained the softmax classifier with three trials and tested the rest two trials from each of 45 hot-ends. The identification score for each hot-end is shown in Figure 12. In the confusion matrix, the lighter the cell, the higher the confidence for identification. Generally, cells in the diagonal line are the lightest, which indicates that the ith hot-end is classified as the correct one. For those few randomly distributed light cells, they are the misclassified hot-ends. In overall, the identification accuracy for 45 hot-ends is about 92%. To investigate the effect of the sampling rate of temperature readings on performance, we test the accuracy based on four different sampling frequencies (i.e., 10 Hz, 15 Hz, 25 Hz, and 50 Hz), as shown in Fig. 13. Consistent with our expectation, when the sampling rate increases, the more detailed and subtle temperature variations can be captured, which result in the higher accuracy.

To further evaluate our system while introducing alien hot-ends, we randomly selected 20 hot-ends as the trained classes, and each time we trained one hot-end's temperature traces as the genuine class and the remaining 19 hot-ends' temperature traces as the outsider classes. Based on the classification score of the validation results, we optimized the threshold for the classification score by computing the Equal Error Rate (EER). The receiver operating characteristic (ROC) curve is shown in Figure 14 with an EER = 8.72%. For the testing part, we tested both the 20 trained classes and the other 25 untrained hot-ends (as the alien classes). As shown in Fig. 15, when the number of alien hot-ends is very small (from 1 to 5), the randomness of the intrinsic physical characteristics of each individual hot-end unit would bring high variance of the FAR performance. However, the average FAR remains stable with FAR = 8.72%, when increasing the number of alien hot-ends. It is shown that, even for unknown hot-ends, our model is suitable and effective for large-scale representation.

D. System Robustness

1) Variations of Room Temperature: Room temperature is a significant and common factor that may affect the temperature readings on 3D printers. Also, in the real environment, we cannot guarantee that every trial was performed with the identical room temperature. To reduce the influence of room temperature variations, we normalized all temperature reading curves by setting a standard temperature start point that higher than the room temperature, such as 30 degrees Celsius, and recording 1,500 samples starting from this standard temperature start point.

2) Permanence of 3D Printer Fingerprints: Like conventional biometrics, permanence (i.e., performance consistency

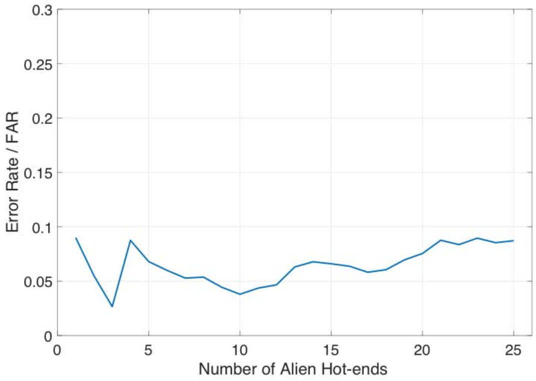


Fig. 15. Error rate for alien hot-ends.

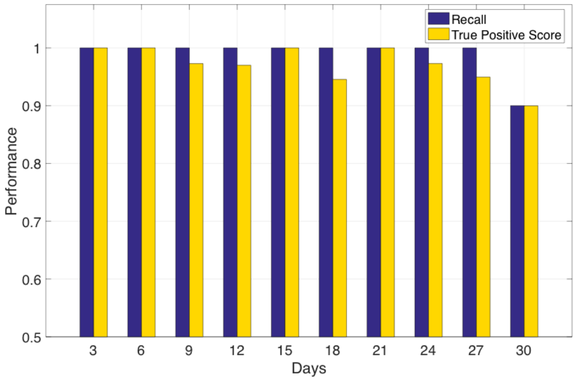


Fig. 16. Performance of recall rate (in blue) and true positive score (in yellow) for the same hot-end over a duration of up to one month.

over time) is another important factor for hardware fingerprinting. Although the best usage metric is the exact working hours, considering the target 3D printer is in normal daily usage, we thus take the calendar days to approximate the usage level. We randomly selected one hot-end and kept recording the hot-end's temperature readings of pre-heating over one month and found that the temperature reading curves indeed had some variations compared with the original one. To present these variations, we used the true positive verification score from the classifier as the evaluation parameter. For up to one-month duration, we measured the hot-end's fingerprint on 10 testing days with 3 days interval and repeated 10 times on each testing day. The result is shown in Figure 16, the recall rate remains 100% for the first 27 days and drops 10% only on the last day. To better present the variance, we also list the classifier scores which can be considered as the probability to be labeled as true positive. It is observed that the classifier scores gradually decrease, although still remaining at relatively high levels. The graceful performance degradation would be largely attributed to the slight change of thermodynamic properties of the hot-end as a result of the reduced heating efficiency of the cartridge heater and accumulation of filament residue in the nozzle.

VII. ThermoTag-Based Watermarking

The ThermoTag fingerprints extracted from our proposed scheme can be used for watermarking 3D-printed products and protecting the IPs of the genuine 3D design. This section will give an example of utilizing layer deformation as the watermarking process in real scenarios.

A. Watermark Generation

Inspired by the watermarking methods specifically for FDM 3D printing [54], [55], we designed a watermarking scheme to embed the ThermoTag of a 3D printer into the 3D-printed objects. This method aims to encode each ThermoTag number into a 12-bit binary number. As shown in Fig. 17, the ThermoTag is the outputs of the hidden layer with the sigmoid activation function which ranges from 0 to 1. To check the integrity of the ThermoTag, an adapted ISBN10 checksum is firstly used, which is embedded as a new bit of the watermark in every 10th layer (5 layers in the Fig. 17 as an example), defined as the follows:

$$p_{10} = ((\sum_{i=1}^{9} i * p_i) \mod 7) + 1 \tag{31}$$

where p_i is the ThermoTag number in the ith layer, p_{10} is the checksum number. Moreover, to identify the degree information of the top and bottom for decoding, the initial two layers are reserved as the reference points.

B. Watermark Embedding

1) Watermark Pattern: To embed an M layers' ThermoTag with the checksum number watermark, after adding the checksum and initial numbers, we then convert them (floating numbers) into binary numbers. As shown in Fig. 18, it shows the encoding of one layer of binary numbers by locally modifying the thickness of the two printing layers. The layers are divided into equally sized encoding and seperating regions for each bit. In the encoding regions of each bit, the thickness of the bottom layers is multiplied by the factor (1+α) or (1-α) to encode the bit 1 or 0, respectively. The top layer thickness is adjusted to keep the sum of the two layers' thickness. An example of the encoded pattern is shown in Fig. 19 with embedded watermark bits "1" and "0". To avoid material accumulated deformation and increase decoding resolution, each encoding layer is separated by M "N/A" layer with normal layer thickness.

In practice, we use the settings of $\alpha = 0.3$, $M = 2$, $\text{Coding}_{\text{width}} = 1.2$ mm, and $\text{Gap}_{\text{width}} = 0.6$ mm. Considering the printing and scanning errors, to ensure the accuracy of expanded watermark bits, we further introduce a local correlation check mechanism [55]. Specifically, we check the sums of binary watermarks in the corresponding layer and in the corresponding column, respectively, which largely reduce the printing and scanning errors. For example, if the watermark "1" indicated by the red box in Fig. 17 is accidentally recognized as "0", we can localize the error by checking the corresponding layer-sum and column-sum.

2) Pattern Printing: In order to modify the layer thickness, we precisely adjust the PLA filament volume extruded from the hot-end during the printing process proposed in [56]. We obtain the volume of the PLA filament by multiplying the

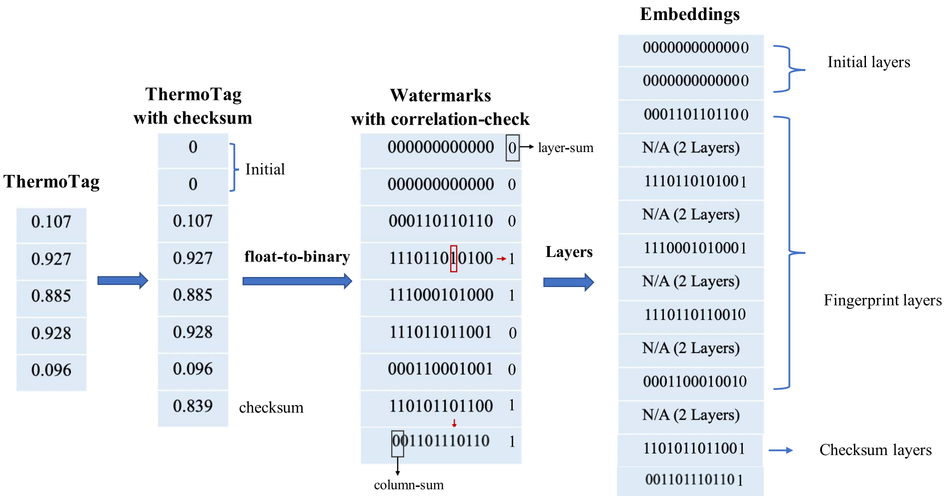


Fig. 17. An example of the watermark generation and encoding steps (5 ThermoTag number).

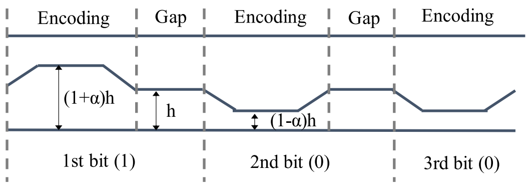


Fig. 18. Encoding layer pattern. The pattern corresponds to two layers with variable thickness.

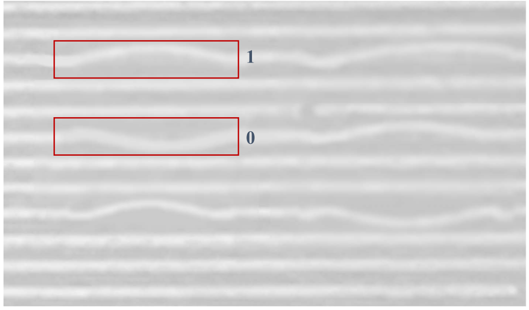


Fig. 19. An example of the encoding layer pattern. Red boxes indicate the embedded watermark bits "1" and "0".

cross-sectional area by the length of the layer, and calculate the filament length, as shown below:

$$L_{\text{filament}} = \frac{A_{\text{layer}} L_{\text{layer}}}{\pi \left(\phi_{\text{filament}}/2\right)^2} \tag{32}$$

where A_{layer} is the cross-sectional area, L_{filament} is the extruded filament length, ϕ_{filament} is the filament diameter, and L_{layer} is the length of the layer.

C. Watermark Decoding

1) Pattern Detection: To detect the embedded watermarks, we first align the scanned image by utilizing Radon transform [57]. Then, we detect the peak magnitude value of the Radon transform over all angles, compute the corresponding rotation, and reorient the image. After that, we recognize the edges between the encoding layers and separating layers using the Canny filter which is a robust edge detection algorithm [58], as shown in Fig. 20. Then we analyze the edges information between the encoding layers in the range between the top half of the bottom encoding layer and the bottom half of the top encoding layer shown in Fig. 21, which restricts the searching region for these edges and is robust to printing noises. We firstly find the largest distance gap between the encoding layer and bottom half of the top layer h_1 and top half of the bottom layer h_2. Then based on the peak point, we can calculate the watermark encoding region with the fixed width d. After every edge has been robustly detected, we can decode the value of each watermark bit by analyzing the layer thickness of encoding layers.

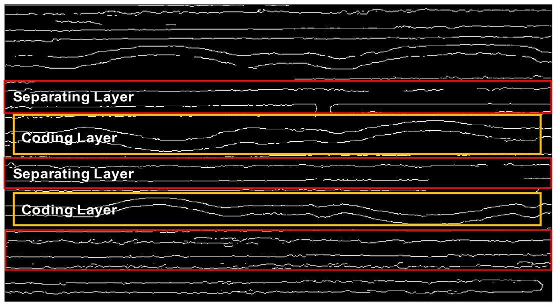


Fig. 20. Decode the printing pattern of separating layer and coding layer by using the edge detection.

2) Watermark Verification: After decoding the pattern, we first check each watermark row and column based on the layer-sum and column-sum bits and localize the possible wrong watermark bit. The correlated watermarks are converted from the binary format into ThermoTag in the format of floating-point numbers. We then check the integrity based on the checksum bit from Equation (31). Finally, we match the decoded ThermoTag with our stored template to verify the watermark.

D. Proof-of-Concept Case Study

As described in Section V-C, to validate the effectiveness of the proposed ThermoTag-based watermarking scheme, we

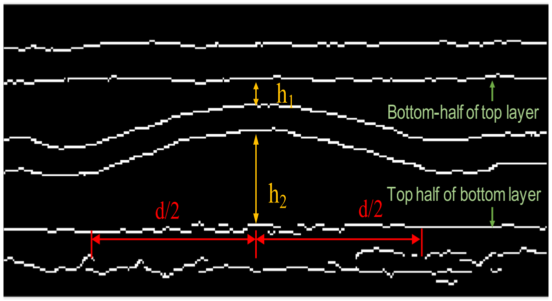


Fig. 21. Watermark decoding by finding the largest difference of distances ($|h_1 - h_2|$) between the encoding layer and top and bottom layers, respectively.

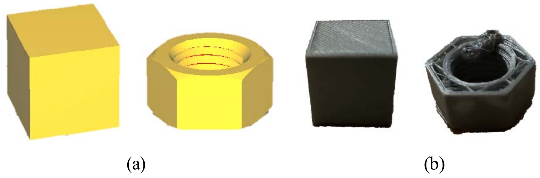


Fig. 22. Printed models in the experiment "cube" and "hexagonal nut". (a) CAD models; (b) actual printed models.

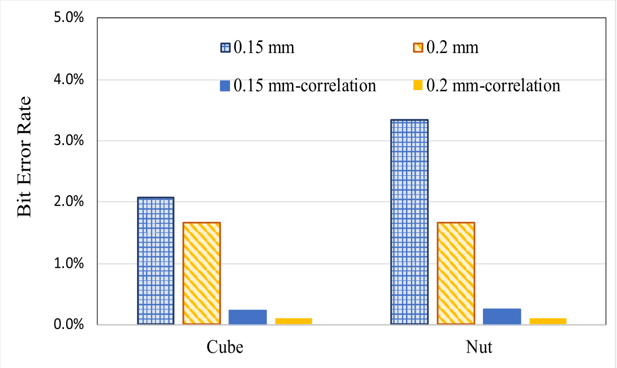


Fig. 23. Bit error rate for two printed models "cube" and " hexagonal nut" under different layer thickness settings.

first extract the ThermoTag which is a set of intrinsic features from the temperature curves through a sparse auto-encoder model. As a proof of concept, a 9-bit watermark is generated based on the ThermoTag, and then encoded into different layers of the 3D design with modified layer thickness in "G-code" sliced by "Ultimaker Cura". As shown in Fig. 22, we choose two different models in our experiment, a standard cube (20 mm × 20 mm × 20 mm) and a hexagonal nut with inside screw thread (24 mm × 27 mm × 13 mm). When the single wall cannot hold the entire binary watermark sequence, we print the remaining portion of the watermark sequence on the next side of the wall. To evaluate the effectiveness of watermarks on different layer thickness, we printed five times for each model under different layer thickness settings. In total, we printed ten times for each model with embedded watermarks. We recorded the side-view images of the watermark patterns by a camera with the resolution of 12 megapixels. The result is shown in Fig. 23. It is observed that the correlation-check could greatly reduce the error rates for both layer settings, and the bit error rates for 0.2 mm are generally lower than the error rates for 0.15 mm. The reason could be that the extrusion variations or accumulated extrusion distortions during printing have higher impacts on thinner layers.

VIII. Discussion and Conclusion

This study proposes a new hardware fingerprinting approach for emerging 3D printers, leveraging the uniqueness of thermodynamic characteristics of the hot-ends in 3D printers, to prevent against counterfeiting and cyber-physical attacks. The hot-end-based ThermoTag fingerprints can be easily obtained through temperature sensing and used as the 3D printing watermarks, safer than using extrinsic elements or random numbers, and more efficient than introducing PUFs into 3D printed products. The unique features of hot-ends are caused by the manufacturing imperfections and system variations, which are hard to identify, predict, and clone. The investigation on 3D printer fingerprinting is still in the infant stage but holds great potential for IP protection or forensics in 3D printing.

A. Environmental Factors

Room temperature and time are not the only two environmental factors. After printing for an extended period of time, the melted filament may coat the nozzle, which is very hard to remove completely and will gradually change the thermodynamic properties of the hot-ends. In addition, the moisture level is another factor that might affect the heat convection in the air, and thus affect the temperature readings as well. All these changes caused by the environment or regular daily usage indicate that for authentication purposes, the hot-end-based 3D printer fingerprints in the database need to be periodically calibrated and updated.

B. Multi-Level Fingerprinting

We have demonstrated that the temperature readings during the pre-heating process can provide unique features of 3D printer hot-ends. From our preliminary experiments, we have observed that standalone printers will have different responses to the same printing command during the printing. Thus, instead of solely based on the unique thermal behavior of pre-heating, combining the other printing behaviors such as temperature or motor oscillation with our proposed fingerprint, could further increase the accuracy and robustness of discrimination.

C. 3D Watermarking

In this work, we utilized the layer-deformation approach to encode the ThermoTag into 3D-printed objects. It still holds the potential risk of being noticed or modified by attackers. In the future work, to help further provide a more obscure and secure fingerprint, a 3D mesh-model based blind watermark technique (e.g., [46], [59]) could be included in the encoding scheme.

References

[1] I. Campbell, O. Diegel, R. Huff, J. Kowen, and T. Wohlers, "Additive manufacturing and 3D printing state of the industry," Wohlers Associates, Fort Collins, CO, USA, Wohlers Rep. 2019, 2019.

[2] Z. Li, A. S. Rathore, C. Song, S. Wei, Y. Wang, and W. Xu, "PrinTracker: Fingerprinting 3D printers using commodity scanners," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 1306–1323.

[3] B. Berman, "3D printing: The new industrial revolution," *IEEE Eng. Manag. Rev.*, vol. 41, no. 4, pp. 72–80, Dec. 2013.

[4] W. Gao *et al.*, "The status, challenges, and future of additive manufacturing in engineering," *Comput.-Aided Des.*, vol. 69, pp. 65–89, Dec. 2015.

[5] E. Macdonald *et al.*, "3D printing for the rapid prototyping of structural electronics," *IEEE Access*, vol. 2, pp. 234–242, Dec. 2014.

[6] C. L. Ventola, "Medical applications for 3D printing: Current and projected uses," *Pharmacy Therapeutics*, vol. 39, no. 10, p. 704, 2014.

[7] SpaceX. (2014). *SpaceX Lauches 3D-Printed Part to Space, Creates Printed Engine Chamber*. Accessed: Apr. 1, 2019. [Online]. Available: http://www.spacex.com/news/2014/07/31/spacex-launches-3d-printed-part-space-creates-printed-engine-chamber-crewed/

[8] D. Sheynin and Y. Bovalino. (2017). *A Treat for the AvGeeks: An Inside Look at GE's 3D-Printed Aircraft Engine*. Accessed: Apr. 1, 2019. [Online]. Available: https://www.ge.com/reports/treat-avgeeks-inside-look-ges-3d-printed-aircraft-engine/

[9] A. Liptak. (Jul. 29, 2017). *The US Navy 3D Printed a Concept Submersible in Four Weeks*. Accessed: Apr. 1, 2019. [Online]. Available: https://www.theverge.com/2017/7/29/16062608/us-navy-3d-printing-submersible-manufacturing-military

[10] G. C. Anzalone, C. Zhang, B. Wijnen, P. G. Sanders, and J. M. Pearce, "A low-cost open-source metal 3-D printer," *IEEE Access*, vol. 1, pp. 803–810, Dec. 2013.

[11] M. Weinberg, *What's the Deal With Copyright and 3D Printing?* Washington, DC, USA: Public Knowledge, 2013.

[12] 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.

[13] L. Sturm, C. Williams, J. Camelio, J. White, and R. Parker, "Cyber-physical vulnerabilities in additive manufacturing systems," *Context*, vol. 7, no. 8, pp. 951–963, 2014.

[14] 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.

[15] Y. Gao, B. Li, W. Wang, W. Xu, C. Zhou, and Z. Jin, "Watching and safeguarding your 3D printer: Online process monitoring against cyber-physical attacks," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, p. 108, 2018.

[16] S. R. Chhetri, A. Barua, S. Faezi, F. Regazzoni, A. Canedo, and M. A. Al Faruque, "Tool of spies: Leaking your IP by altering the 3D printer compiler," *IEEE Trans. Dependable Secure Comput.*, early access, Jun. 20, 2019, doi: 10.1109/TDSC.2019.2923215.

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

[18] J.-U. Hou, D.-G. Kim, S. Choi, and H.-K. Lee, "3D print-scan resilient watermarking using a histogram-based circular shift coding structure," in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2015, pp. 115–121.

[19] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1999, pp. 49–56.

[20] R. Arppe and T. J. Sørensen, "Physical unclonable functions generated through chemical methods for anti-counterfeiting," *Nature Rev. Chem.*, vol. 1, no. 4, pp. 1–13, Apr. 2017.

[21] N. Gupta, F. Chen, N. G. Tsoutsos, and M. Maniatakos, "ObfusCADe: Obfuscating additive manufacturing CAD models against counterfeiting: Invited," in *Proc. 54th Annu. Design Automat. Conf.*, Jun. 2017, pp. 1–6.

[22] S. Yamazaki, S. Kagami, and M. Mochimaru, "Extracting watermark from 3D prints," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 4576–4581.

[23] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device identification via sensor fingerprinting," 2014, *arXiv:1408.1416*. [Online]. Available: http://arxiv.org/abs/1408.1416

[24] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "AccelPrint: Imperfections of accelerometers make smartphones trackable," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014, pp. 1–16.

[25] J. F. Sargent, Jr., and R. X. Schwartz, "3D printing: Overview, impacts, and the federal role," Congressional Res. Service, Washington, DC, USA, Tech. Rep. R45852, Aug. 2019.

[26] J. M. Pearce, C. M. Blair, K. J. Laciak, R. Andrews, A. Nosrat, and I. Zelenika-Zovko, "3-D printing of open source appropriate technologies for self-directed sustainable development," *J. Sustain. Develop.*, vol. 3, no. 4, pp. 17–29, Nov. 2010.

[27] R. Jones *et al.*, "RepRap—The replicating rapid prototyper," *Robotica*, vol. 29, no. 1, pp. 177–191, 2011.

[28] T. Greene, "U.S. 3D printer forecast, 2016–2020: New 3D print/additive manufacturing technologies fuel growth," IDC Res., Framingham, MA, USA, Tech. Rep. US41333516, May 2016.

[29] M. Yampolskiy, T. Andel, J. McDonald, W. Glisson, and A. Yasinsac, "Intellectual property protection in additive layer manufacturing: Requirements for secure outsourcing," in *Proc. Program Protection Reverse Eng. Workshop*, no. 7, 2014, pp. 1–9.

[30] M. A. Al Faruque, S. R. Chhetri, A. Canedo, and J. Wan, "Acoustic side-channel attacks on additive manufacturing systems," in *Proc. ACM/IEEE 7th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2016, pp. 1–10.

[31] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. Al Faruque, "Thermal side-channel forensics in additive manufacturing systems," in *Proc. 7th ACM/IEEE Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2016, p. 22:1.

[32] A. Hojjati *et al.*, "Leave your phone at the door: Side channels that reveal factory floor secrets," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 883–894.

[33] C. Song, F. Lin, Z. Ba, K. Ren, C. Zhou, and W. Xu, "My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3D printers," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 895–907.

[34] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conf. Comput. Commun. Secur. (CCS)*, 2002, pp. 148–160.

[35] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Jun. 2008, pp. 51–57.

[36] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2008, pp. 670–673.

[37] K. D. D. Willis and A. D. Wilson, "Infrastructs: Fabricating information inside physical objects for imaging in the terahertz region," *ACM Trans. Graph.*, vol. 32, no. 4, p. 138, 2013.

[38] O. Ivanova, A. Elliott, T. Campbell, and C. B. Williams, "Unclonable security features for additive manufacturing," *Additive Manuf.*, vols. 1–4, pp. 24–31, Oct. 2014.

[39] H. Cheng *et al.*, "Plasmonic nanopapers: Flexible, stable and sensitive multiplex PUF tags for unclonable anti-counterfeiting applications," *Nanoscale*, vol. 12, no. 17, pp. 9471–9480, 2020.

[40] C. Song, Z. Li, W. Xu, C. Zhou, Z. Jin, and K. Ren, "My smartphone recognizes genuine QR codes!: Practical unclonable QR code via 3D printing," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 2, p. 83, 2018.

[41] F. Peng, J. Yang, Z.-X. Lin, and M. Long, "Source identification of 3D printed objects based on inherent equipment distortion," *Comput. Secur.*, vol. 82, pp. 173–183, May 2019.

[42] B. Macq, P. R. Alface, and M. Montanola, "Applicability of watermarking for intellectual property rights protection in a 3D printing scenario," in *Proc. 20th Int. Conf. 3D Web Technol.*, Jun. 2015, pp. 89–95.

[43] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 5, pp. 596–607, Sep. 2005.

[44] J.-W. Cho, R. Prost, and H.-Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 142–155, Jan. 2007.

[45] Y. Yang, R. Pintus, H. Rushmeier, and I. Ivrissimtzis, "A 3D steganalytic algorithm and steganalysis-resistant watermarking," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 2, pp. 1002–1013, Feb. 2017.

[46] J.-U. Hou, D.-G. Kim, and H.-K. Lee, "Blind 3D mesh watermarking for 3D printed model by analyzing layering artifact," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2712–2725, Nov. 2017.

[47] F. Peng, J. Yang, and M. Long, "3-D printed object authentication based on printing noise and digital signature," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 342–353, Mar. 2019.

[48] W. M. Kays, M. E. Crawford, and B. Weigand, *Convective Heat and Mass Transfer*. New York, NY, USA: McGraw-Hill, 2012.

[49] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst., Meas., Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993.

[50] K. J. Åström and T. Hägglund, *Advanced PID Control*. New Delhi, India: International Society of Automation, 2006.

[51] J. Bhatnagar and A. Kumar, "On estimating performance indices for biometric identification," *Pattern Recognit.*, vol. 42, no. 9, pp. 1803–1815, Sep. 2009.

[52] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[53] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.

[54] F. W. Baumann and D. Roller, "Watermarking for fused deposition modeling by seam placement," in *Proc. MATEC Web Conf.*, vol. 104, 2017, p. 02023.

[55] A. Delmotte, K. Tanaka, H. Kubo, T. Funatomi, and Y. Mukaigawa, "Blind watermarking for 3-D printed objects by locally modifying layer thickness," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2780–2791, Nov. 2020.

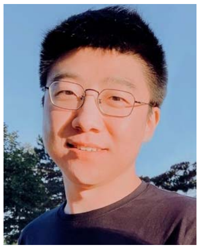
[56] G. P. Greeff and M. Schilling, "Closed loop control of slippage during filament transport in molten material extrusion," *Additive Manuf.*, vol. 14, pp. 31–38, Mar. 2017.

[57] K. Jafari-Khouzani and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 1004–1008, Jun. 2005.

[58] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

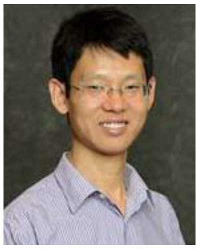
[59] M. R. Mouhamed, M. M. Solima, A. A. Darwish, and A. E. Hassanien, "Robust and blind watermark to protect 3D mesh models against connectivity attacks," in *Proc. 8th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2017, pp. 23–29.

Yang Gao received the B.S. degree in microelectronics from the University of Electronic Science and Technology of China in 2012 and the M.S. degree in electrical engineering from the Stevens Institute of Technology in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering with the University at Buffalo, The State University of New York (SUNY). His research interests include biometric-based user identity verification, the IoT sensing and cyber-physical security, mobile and wearable computing, and machine learning.



Wei Wang received the B.S. degree in automation from Northwestern Polytechnical University, China, in 2008. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York (SUNY). Prior to this, he was a Ph.D. candidate with the Department of Electrical and Computer Engineering, Binghamton University, SUNY. His research interests include deep learning, mobile sensing, human–computer interaction, and mobile and wearable computing.

Yincheng Jin received the B.S. degree in microelectronics from Northwestern Polytechnical University. He is currently pursuing the Ph.D. degree in computer science and engineering with the University at Buffalo, The State University of New York (SUNY). His research interests include the IoT sensing, mobile and wearable computing, and machine learning.



Chi Zhou received the master's degree in computer science from USC in 2010 and the Ph.D. degree in industrial and systems engineering from the University of Southern California in 2012. He is currently an Associate Professor with the Department of Industrial and Systems Engineering, University at Buffalo. Prior to joining UB in July 2013, he was a Senior Research and Development Engineer with EnvisionTec Inc. His current research interests include computer-aided design and manufacturing (CAD/CAM) related to direct digital manufacturing.

Wenyao Xu (Senior Member, IEEE) received the bachelor's and master's degrees from Zhejiang University, China, and the Ph.D. degree from the University of California at Los Angeles, Los Angeles, USA. He is currently an Associate Professor with the Department of Computer Science and Engineering, University at Buffalo (SUNY). His research has focused on exploring novel sensing and computing technologies to build up the innovative Internet-of-Things (IoT) systems for high-impact human-technology applications in the fields of smart health and cyber-security. Results have been published in peer-reviewed top research venues across multiple disciplines, including Computer Science conferences (e.g., ACM MobiCom, SenSys, MobiSys, UbiComp, ASPLOS, ISCA, HPCA, Oakland, NDSS, and CCS), Biomedical Engineering journals (e.g., IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING (TBME), IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS (TBIOCAS), and IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (JBHI)), and Medicine journals (e.g., LANCET). His group has published more than 190 peer-reviewed articles, won 14 Best Paper Awards and nominations in top conferences (e.g., ACM MobiSys, ACM SenSys, and IEEE BHI), and three international Best Design Awards. His inventions have been filed within the U.S. and internationally as patents and have been licensed to industrial players. His research has been reported in high-impact media outlets, including the Discovery Channel, CNN, NPR, and the Wall Street Journal. He has been the TPC Co-Chair of 2018 IEEE Body Sensor Networks. He serves as an Associate Editor for IEEE INTERNET-OF-THINGS JOURNAL (IOTJ) and IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS (TBCAS). He serves on the technical program committee for numerous conferences in the field of mobile computing, smart health, and the Internet of Things.

Zhanpeng Jin (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and engineering from Northwestern Polytechnical University and the Ph.D. degree in electrical engineering from the University of Pittsburgh. He is currently an Associate Professor of computer science and engineering with the University at Buffalo, The State University of New York (SUNY). His research interests include emerging biometrics, mobile and wearable computing, the IoTs and cyber-physical systems, ubiquitous sensing, and smart health. He has served as an Associate Editor for the following journals: IEEE ACCESS, *Computers in Biology and Medicine*, and *BioMedical Engineering OnLine*.