Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

NFTs For 3D Models: Sustaining Ownership In Industry 4.0

Dimitris Mouris and Nektarios Georgios Tsoutsos
Department of Electrical and Computer Engineering,
University of Delaware, Newark, DE, 19716, USA

Digital manufacturing (DM) is actively adopted to the production lifecycles of a variety of critical industries, and this rapid growth has resulted in exponential increase of 3D computer-aided design (CAD) models. Unfortunately, counterfeiting of intellectual property becomes a prominent threat as many 3D designs are accessible online, combined with the proliferation of cheap consumer 3D printers that enable malicious actors to produce non-authentic parts. State-of-the-art techniques to secure manufacturing processes mostly rely on watermarking, which embeds hidden information inside CAD models to prove ownership and authenticity. Nevertheless, such techniques tamper with the model itself, while existing attacks allow removing such watermarks altogether.

To address these shortcomings, we integrate signal processing and cryptographic techniques and describe a tailored solution for CAD model ownership and supply chain management. Our approach generates unique identifiers for 3D designs using frequency-domain transforms and employs non-fungible tokens (NFTs) that persist on public distributed ledgers. Our NFTs are implemented on the Ethereum blockchain using smart contracts and their functionality is twofold: (a) authenticate the owner of a CAD model, and (b) enable ownership transfer. To validate our technique, we deployed our smart contract on Ethereum's proof-of-work Ropsten network and demonstrated the applicability of our methodology.

Index Terms - 3D Printing, Blockchain, Digital Manufacturing, IP Theft, Non-Fungible Tokens, Smart Contracts.

This work was supported by the National Science Foundation under Grant CMMI-1931916.

I. IP Challenges For 3D Computer-Aided Designs

THE fourth industrial revolution, also called *Industry 4.0*, is characterized by intelligent interconnected systems that exchange information and automate production and supply chain management. Digital manufacturing (DM), including 3D printing, represents one of the fundamental technologies of Industry 4.0 and has been adopted by a variety of sectors including manufacturing, aerospace, and medical due to the continuously improving quality and flexibility of manufactured parts [1]–[3]. The increasing adoption of DM by different industries results in a rapid expansion of 3D computer-aided design (CAD) models.

Nevertheless, this accelerated growth brings new challenges: a major concern in DM is the protection of intellectual prop-erty (IP) against piracy and counterfeiting [4]. For example, developing competing products with inferior materials, modifying stolen designs, or overproducing unauthorized parts can significantly impact the business models of Industry 4.0 [5]. Indeed, when the stolen digital files are used to produce parts of equivalent quality to the original ones, the malicious actors incur significant financial losses for the original IP owner. Therefore, it is imperative to investigate novel methodologies for trustworthy CAD models to mitigate this class of attacks. Recent examples to enable copyright protection include physical watermarks, QR codes, and fingerprints in the 3D models [6]. These methods rely on hidden information embedded into the 3D files, such as the designer's signature, which enables claiming ownership of the original file or tracking the source of piracy by embedding the buyer's signature. All these techniques, however, inevitably alter the original 3D files and are also susceptible to a variety of attack vectors [7], [8].

One promising solution to track DM artifacts and record transactions is the use of distributed ledgers [9]. In 2009, blockchain technology enabled a new form of digital currency that operates solely in decentralized computer nodes. While Bitcoin was the first cryptocurrency based on blockchain [10], many alternative protocols and digital currencies have emerged with Ethereum being one of the most popular [11]. In addition to public ledger properties (e.g., transparency and information immutability), Ethereum also supports event-driven blockchain transactions called *smart contracts*. Smart contracts execute custom business logic based on a public trigger, such as a specific blockchain transaction, and are honored by all participating nodes without the need of trusted third parties. Today, smart contracts enable many applications, from automated payments based on public events to online insurance, decentralized auctions, and even energy trading [12], [13].

Recent advancements in Ethereum's protocol offer a novel functionality: the ability to prove ownership of Non-Fungible Tokens (NFTs), which are *unique digital artifacts* that are stored on the blockchain. Contrary to cryptocurrency coins that are fungible (e.g., any Bitcoin is equivalent to any other Bitcoin), each NFT is one-of-a-kind and can represent a variety of physical or digital assets. Prominent applications include collectibles, digital art, music, video game items, or even real estate, where each NFT acts as cryptographic proof of ownership of the corresponding asset. For example, in the real estate scenario, an NFT corresponds to all relevant legal evidence (including reports, disclosures, and images of the property) so that owning the NFT indicates ownership of the real-world property. Notably, the immutability property of NFTs renders them ideal for expressing ownership and authenticity of assets stored on a blockchain, ensuring that it is impossible to clone the digital artifact for a given asset.

One potential limitation of NFTs, as with any blockchain-based technology, is the inherent cost of publishing large amounts of data on the distributed ledger. Thus, only the very essential information is stored on-chain (e.g., a token identifier, the identity of the owner, and a URL address pointing to the asset), while the rest is stored as off-chain metadata either on a

Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

centralized server or on the InterPlanetary File System (IPFS). The latter is a decentralized peer-to-peer file system that allows replication of files across numerous different locations, while providing an affordable and reliable storage solution. In this case, applications can ensure the immutability of the corresponding IPFS files (pointed by the URL in the NFT) by authenticating their unique hash digest as on-chain data.

Our work explores a new methodology for proving ownership of 3D CAD models based on signal processing, NFTs, and distributed ledger technology. The novelty of our proposed technique is twofold: a) we store 3D design identifiers on a blockchain network to preserve the IP rights of the owner (and therefore we utilize the security of blockchain networks), and b) we combine on-chain information with signal-processing techniques to maintain the IP of the model even under a series of modifications on the CAD model: for instance, changing a small part of the 3D design is still detectable in our approach. Our NFTs are connected to frequency-domain representation of the *shape of CAD models* instead of solely a hash digest of the file itself, which makes our method resilient to small modifications of the file. Specifically, our technique examines the object's silhouette, instead of simply comparing file hashes, and leveraging such shape information it creates immutable fingerprints that are used to authenticate each CAD model. Another key contribution of our work is that it does not tamper with the IP since it only utilizes unique frequency-domain transformation. Building upon the properties of blockchain and smart contracts, such as automation and transparency, our NFTs for 3D designs offer a new way of proving IP ownership in Industry 4.0. At the same time, the proposed method can ensure the integrity of the supply chain by offering cryptographically secure asset tracking using a publicly available transcript of all the transactions and actions applied to a given CAD model on the blockchain ledger.

II. Key Insights

A. 3D CAD Model Recognition Using Spectrograms

Recognition and search in 3D models have been made possible by leveraging multi-dimensional Fast Fourier Transforms (FFTs) to create specialized spectrograms and extract magnitude peak patterns in the frequency domain [14], [15]. A spectrogram represents signal amplitudes as a function of frequency and another variable such as time or distance on a specific axis. In this case, shape information is encoded as a superposition of frequencies and does not depend on the individual 3D CAD file format. Nevertheless, since there exist various file formats used by CAD software (and many of them are proprietary), without loss of generality we focus on the Standard Triangle Language (STL) file format that is universal and can be read by all 3D printers and CAD programs. STL files describe the exterior surface of 3D models as a closed set of interconnected facets (small triangles).

In Fig. 1 we summarize the shape-based search methodology of the Fourier Fingerprint Search (FFS) framework [14]. First, FFS slices the given 3D models across all axes and projects each three-dimensional slice onto a 2D grid, as shown in Fig. 1 (c). Then, each 2D grid projection is translated into the frequency domain using multi-dimensional FFT to create a spectrogram (Fig. 1 (d)), before filtering the location of the highest magnitude peaks (Fig. 1 (e)). Finally, FFS analyzes the magnitude peak patterns to create unique identifiers, dubbed *signatures*, that correspond to the shape of the 3D model (Fig. 1 (f)); these signatures enable searching for similar models in a database.

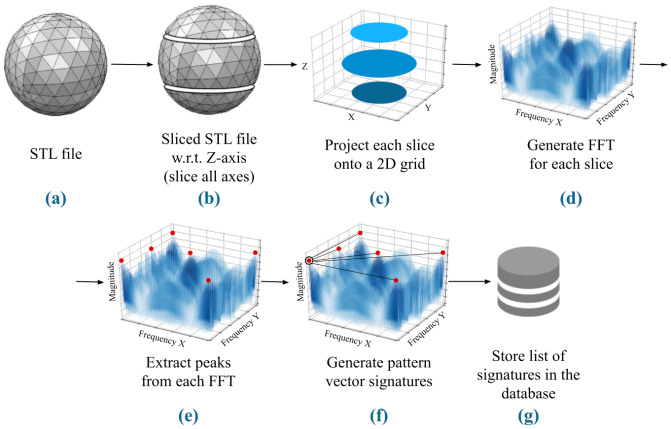

Fig. 1. Overview of Fourier Fingerprint Search (FFS) for 3D models: the framework employs frequency spectrograms to search for patterns that indicate similar shape characteristics.

FFS introduces many optimizations to improve its search functionality: the most prominent optimizations involve rotating the 3D model across different angles to generate additional signatures based on the *silhouette of the model*. These rotations reveal additional features of the object and therefore improve the matching accuracy. Using signature similarity metrics, FFS can search for both exact and partial matches: for a given input, FFS retrieves models that are almost identical to the query, as well as models that share smaller portions with the queried model. This flexibility stems from the fact that all signatures of the 3D model are generated from relatively thin slices, which enables searching for similar files based on small feature differences. As reported in [14], FFS achieves up to 100% average top-5 accuracy while matching an altered CAD file with its original source, based on 3000 CAD designs from the Fabwave dataset. Interestingly, even if the query models have been tampered in a variety of ways, FFS is still able to detect partial matches. More specifically, [14] achieves very high accuracy even after a series of modifications such as: a) degrading the facet resolution by reducing the number of triangles of the STL model, b) introducing perturbations in the CAD file (e.g., altered order of facets), and c) performing random rotations across all axes.

B. Non-Fungible Tokens On a Public Network

A Non-Fungible Token (NFT) is a unique piece of data that is stored on a blockchain to certify that a digital asset, such as a photo, a video, or any other form of a digital file, is unique. Fungible tokens, like cryptocurrency coins, are always interchangeable: each coin is equivalent to any other coin. Conversely, NFTs resemble one-of-a-kind artworks that are authentic and cannot be replicated (e.g., there is only

Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

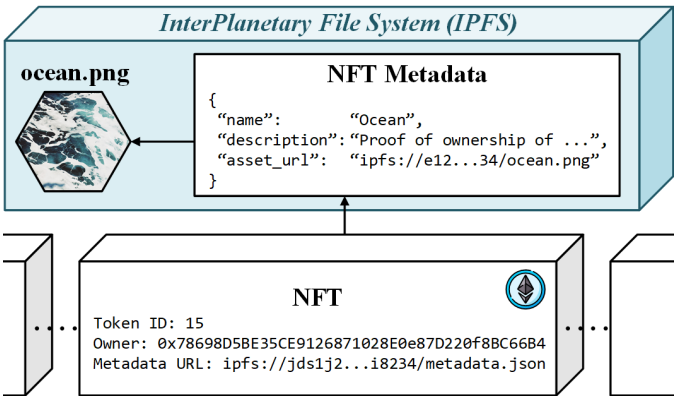

Fig. 2. Example of a non-fungible token representing ownership of a digital artwork. The actual image and the NFT metadata are stored off-chain in the interplanetary file system (IPFS), while the unique token identifier, the owner address and the URL to the metadata file are stored on-chain.

one Mona Lisa). While most NFTs are part of the Ethereum blockchain [11], other blockchain networks are gradually introducing NFT support, and the list is expected to grow. Typically, NFTs employ blockchain records to publicly store and track the ownership of digital assets, by linking a unique identifier of the asset (e.g., its hash digest) with the owner identity (e.g., a public key) in an immutable pair.

Today, most NFTs are focused on buying and selling digital art and digital collectibles; however, combining NFTs with smart contracts enables a variety of real-world applications such as real-estate auctions, copyrights of intellectual properties, or tokenized tickets for events, just to name a few. While digital collectibles can never substitute physical artifacts, the continuous market growth of NFTs reflects their popularity: for instance, in the first quarter of 2021 NFT sales reached $2 billion [16]. Moreover, a crucial benefit of NFTs for digital art over physical artifacts is support for (decentralized) royalty deals, where a fee is collected by the initial creator of a digital collectible every time its NFT is sold to a new owner. As developers of smart contracts are allowed to implement any arbitrary logic, NFTs can support special rules for transferring ownership, or enable unique features for rewarding the original creators.

NFT Standards: The standard for creating and maintaining NFTs using Ethereum smart contracts is governed by the ERC-721 standard (i.e., Ethereum Request for Comments) that was proposed in 2018 [17]. The standard defines how each NFT is cryptographically tied to a distinct identifier that is unique to each owner; for example, Fig. 2 illustrates a simple ERC-721 collectible. We observe that only the NFT identifier (e.g., number *15* in Fig. 2), the owner's address, and a URL pointing to external metadata are stored on-chain. In fact, since on-chain storage is a precious resource, a common practice is to store NFT metadata or even the digital asset itself on a distributed storage resource such as the InterPlanetary File System (IPFS), which provides a secure, transparent, and public way to host immutable data; by design, modifying an IPFS file would inevitably mutate its URL. As illustrated in Fig. 2 (top half), NFT metadata may include the asset name, a brief description, and the IPFS address of the corresponding file (a PNG image in this example).

III. Proving Ownership of 3D Models Using NFTs

The growing number of industries that rely on digital manufacturing renders the authenticity of 3D models a real concern. To address this concern, we propose a new methodology for proving IP ownership and enabling supply chain management of CAD files using NFTs. One limitation of existing NFT applications (e.g., digital art) is that their guarantees rely solely on the hash of the corresponding digital file; while this prevents identical clones, it remains vulnerable to counterfeits that *look very similar* but still have a different hash. For example, altering one pixel of a digital art piece may cause an imperceptible difference, yet the hash will be very different. In this case, manual inspection is required to determine which file is the counterfeit.

Our approach creates tailored fingerprints of 3D models based on the frequency domain representation of the files (Fig. 3): given a 3D model, we employ the FFS methodology (Section II-A) to generate a set of signatures describing its shape information by encoding the patterns of spectrogram peaks. These signatures comprise the *fingerprint* of the 3D object and can be organized in a cryptographic Merkle tree structure as illustrated in Fig. 4. All leaf nodes of a Merkle tree contain FFS signatures, while all intermediate nodes contain a hash of their child nodes; in this case, the tree is built in a bottom-up fashion and the top level is called a *Merkle root*. This data structure design offers an important benefit compared to other techniques (e.g., Bloom filters): when comparing two Merkle trees, the equality of two nodes ensures that all descendant nodes are the same (i.e., equal subtrees). In more detail, even if one bit of a leaf-node is different between two trees, this will significantly change the Merkle root and allow for efficient comparisons. Therefore, it is sufficient to represent a 3D model using only the root hash of a Merkle tree corresponding to the FFS signatures of its shape, whereas other data structures have significantly higher storage requirements. To authenticate the 3D model on the blockchain, we then create an NFT that encodes the owner ID, the Merkle root and the IPFS address pointing to the off-chain metadata (Fig. 3). Specifically, the IPFS metadata includes all intermediate and leaf nodes of the Merkle tree, along with a link to the CAD file itself, which enables publicly-verifiable binding of the CAD file to the NFT. Any third party judge can run FFS to generate shape signatures from the CAD file stored in IPFS, and then verify that the Merkle root in the NFT matches these signatures.

By storing the Merkle root over the signatures of a 3D model, the owner can claim possession of a CAD file that generates these signatures. In this case, proof that the original file has not been modified is derived directly by comparing the Merkle root of the given 3D model in question with the Merkle root stored in the original NFT. At the same time, any modification of the CAD file would yield a different Merkle root that does not match the one stored in the NFT; however, since our method relies on Fourier domain signatures that encode shape information, we can now invoke the FFS

Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

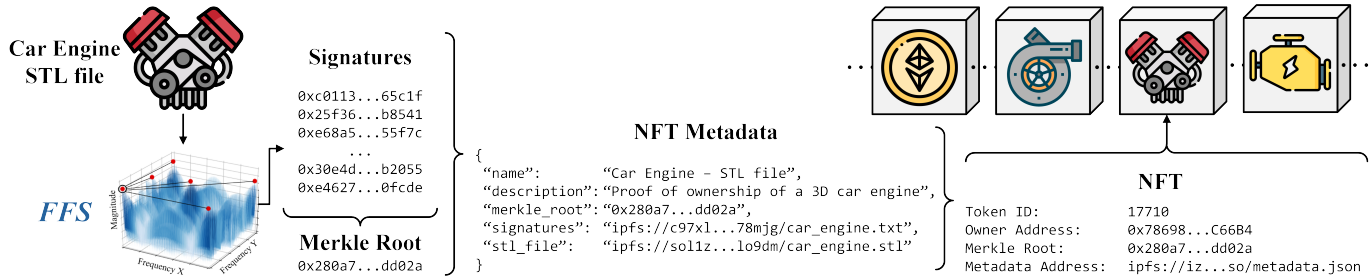

Fig. 3. **Overview of our methodology.** Given an STL file, we utilize the core functionality of FFS to generate numerous signatures and create a Merkle tree based on these signatures. The next step is to upload all the signatures and the STL file to IPFS and then create the *metadata* for the NFT that contains the Merkle root as well as URLs to the signatures and to the STL file. Finally, we mint a new NFT with a unique token identifier, the owner address, the Merkle root, and the metadata URL.

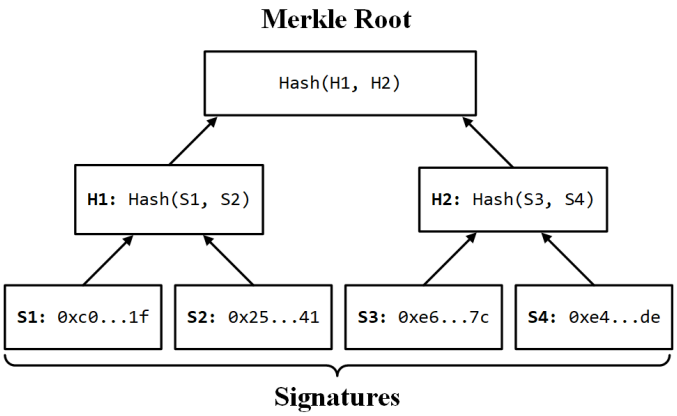

Fig. 4. Merkle Tree construction based on four signatures (i.e., **S1**, **S2**, **S3**, and **S4**). Our NFT methodology stores only the Merkle root.

partial matching routines and compare the query CAD file with the authentic one encoded in the NFT. Notably, this shape comparison method is more valuable when the CAD file contents are different but the shapes look very similar. The latter enables the detection of counterfeit designs that evade existing comparison methods based on file content similarity.

Fig. 5 presents an example of the frequency-based matching of FFS between three similar hex-head bolts (which have different Merkle roots). Bolts (a) and (c) have been created separately and have many different characteristics, whereas bolt (b) is a slightly modified version of bolt (a). We have highlighted the key differences between bolts (a) and (b) in red, and between bolts (a) and (c) in green. The signatures that FFS computes for each of the three bolts in Fig. 5 represent the shape characteristics of the 3D objects and are used to calculate a similarity percentage (often this is different from what humans can perceive); here, bolt (c) has a different thread pattern compared to bolt (a). FFS algorithms report a high similarity factor between bolts (a) and (b) (i.e., about 80%), whereas bolt (c) has a significantly lower similarity factor with either of the other two bolts (e.g., about 15%), indicating that it has been created differently.

A benefit of our methodology is resilience to CAD file transformations, such as rotation by varying degrees across all three axes or reordering of the STL file facets, which generates different files with respect to binary contents. Likewise, degrading the resolution of an object (e.g., combining multiple facets into one) does not significantly impact the frequency representation of its shape, so our methodology remains effective. Thus, our NFT-based ownership technique maintains all the same benefits of watermarking, yet it avoids any modification to the 3D models. Indeed, using FFS routines we can detect counterfeits, while the use of a public ledger on the blockchain authenticates the legitimate owner of the CAD file and help preserve their IP. In this case, the original creator (e.g., the initial owner of the IP) can invoke a smart contract to publicly transfer ownership of the 3D model, which creates a new blockchain transaction indicating the new owner ID for a given NFT.

In our methodology, the FFS routines focus on computing the similarity factor between different models, while the owner can determine a threshold to flag potential counterfeits (subject to further inspection). For example, the owner of the original hex-head bolt in Fig. 5 may indicate that anything below 60% should not be flagged as counterfeit. In the case of more complex 3D models, however, such as a car engine block with a rich set of features, the owner may set a lower threshold (e.g., 20%) in case only part of the design is counterfeited. To increase flexibility, our approach supports variable similarity thresholds defined by each owner for a given model, and these values can be encoded as part of the metadata of the corresponding NFT. At the same time, owners have the option to split complex models into smaller individual parts to facilitate ownership claims; in the previous example of the car engine block, the owner can spread the design across multiple NFTs instead of a single one, which allows proving ownership of the individual parts comprising the engine block. This fine-grained approach is beneficial for complex designs whose individual components may evolve over time.

IV. Instantiation of NFTs for 3D CAD Models

We instantiated a special ERC-721 contract that is tailored to CAD models using *Solidity*, the object-oriented programming language designed for implementing smart contracts. Our Ethereum contract inherits the core functionality of ERC-721 from OpenZeppelin[1] and implements the core functionality for creating bespoke tokens with a unique identifier, a unique

[1]A framework for building smart contracts and ERC standards for Ethereum and other blockchains.

Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

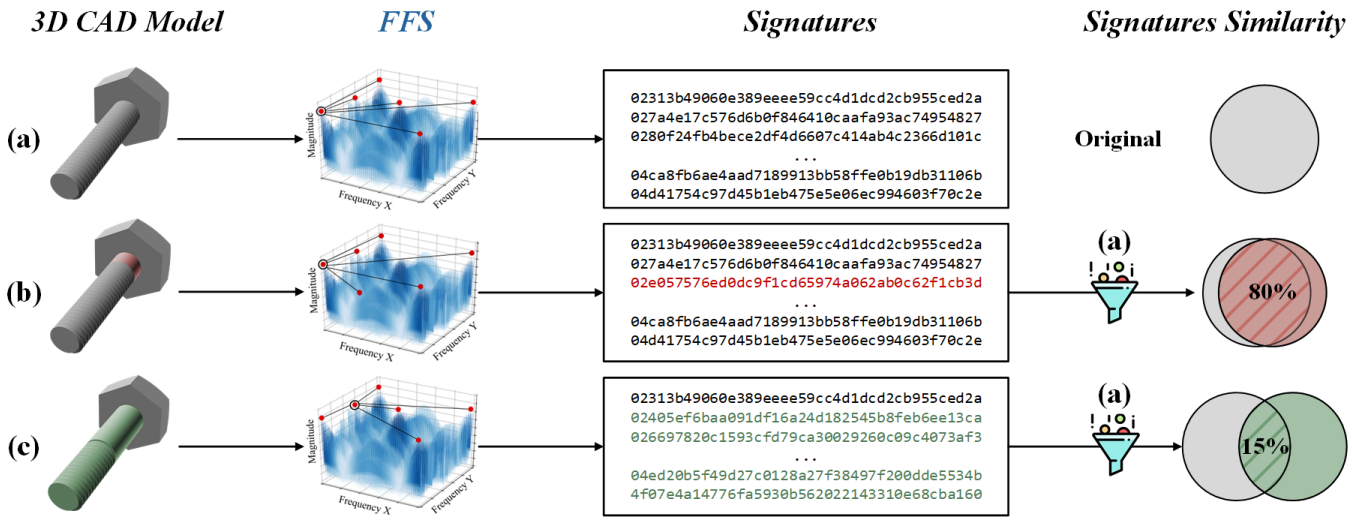

Fig. 5. Example of three similar 3D hex-head bolts. We have highlighted their differences with bolt (a) in red and green color. Bolt (b) is a modified version of bolt (a), whereas, bolt (c) is a different bolt with different thread pattern. Although some of these differences are not always visible to humans, FFS leverages frequency domain characteristics and is able to detect that bolts (a) and (b) incur a high similarity percentage (i.e., they share over 80% of their signatures), while bolt (c) has lower similarity percentages with either of the other two bolts (i.e., below 15%).

owner ID, a metadata URL, and a Merkle root. Our framework also includes a toolchain that automates the NFT creation by integrating signature generation routines from FFS as well as computing the Merkle root and uploading data to IPFS. Overall, the Merkle root computation, along with FFS signature generation for object rotations across all three axes (using six slices), can be executed in about 1 second. FFS was instantiated using Python 3 and LevelDB, an open-source key-value storage library provided by Google.

To interact with the Ethereum network (*Ropsten* public testnet in our case), we employ the Truffle Suite, which is a development environment and testing framework that uses the Ethereum Virtual Machine (EVM). For evaluation, we have deployed our custom smart contract on the blockchain[2] and created a user with four NFTs. Moreover, we employ the `nft.storage` free IPFS service to store our decentralized off-chain information, including the actual STL file, its signatures, and metadata file. For instance, the metadata of *token 4* of our instantiation is uploaded to https://tinyurl.com/bolt-metadata. while the actual NFT resides on https://tinyurl.com/cstl-nft-4.

Every blockchain transaction, such as creating an NFT or deploying a smart contract, has a modest fee for allocating resources to enable smart contract execution on the blockchain. In Ethereum, this fee is called *Gas* and is measured in Gigawei (Gwei) units (equal to 10^{-9} Ethereum coins). When multiple users engage in concurrent transactions, the gas price increases and the transactions with higher bids are prioritized. Moreover, the transaction cost depends on its bytesize, so smaller transactions incur less computational effort and thus less Gas. Therefore, the total fee is based on the current Gas price, as well as required Gas to complete the transaction (i.e., Total = Gas Price * Gas Used). In our evaluation using Ethereum's Ropsten testnet, the gas price was 20 Gwei (0.00000002 Ethereum coins) and creating a new NFT requires 244897 Gas, which corresponds to 0.00489794 coins. Moreover, the deployment cost for our bespoke smart contract is 2,699,306 Gas (0.05398612 coins); we remark that this deployment cost *incurs only once* when we initialize our CAD model authentication service, and it is orthogonal to the NFT creation cost. After deployment, the contract is executed to create a new NFT or enable interaction with a existing one (e.g., charge owner after a sale transaction).

[2]Accessible at https://tinyurl.com/cryptoSTL-contract

V. Recent Works, Open Questions, and Next Steps

A. Defending Against IP Theft Using Watermarks

As AM technology advances, the ability to produce counterfeited artifacts from 3D models is rapidly increasing. One prominent solution is watermarking, a technique that embeds a signature into the model, enabling the original designer to reveal the watermark later and claim ownership. Thwarting IP piracy is of critical importance, however it remains an inherently difficult problem. Common techniques that extrude a signature on the surface of the artifact can be easily bypassed, as attackers can detect and remove the watermark (like removing a watermark from a 2D picture via image editing software). More advanced techniques, such as [18],

TABLE I
Comparisons With Watermarking Protection.

3D CAD IP Protection Technique	Attack Vectors	Model Modification Requirement	Supply Chain Management
Geometry Watermarking [18]	Geometrical transformations, Mesh operations, Compression, simplification, re-reordering, etc.	Embed watermark into geometry.	Extract watermark from geometry and embed new.
Texture Watermarking [19]	Geometrical distortions, Subsampling, Any texture modification.	Embed watermark into texture.	Extract watermark from texture and embed new.
NFT & FFS (this work)	Inherits security of blockchain network.	None.	Natively by smart-contract.

Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

embed a string of bits into the geometrical structure of the object by changing the locations of certain vertices. Such techniques have a significant advantage compared to watermarks on the surface of an object (e.g., [19]), as they hide the watermark *inside* the model, making it more difficult to detect. Unfortunately, even advanced watermarking techniques are subject to removal attacks: For one, regardless of how cleverly the watermark is obfuscated, it can still be found, removed, or even altered [7], [8]. Likewise, the watermark may be corrupted unintentionally during common processing of the model, such as compression. Conversely, our NFT-based solution has two significant benefits: (a) it cannot be removed or altered under any circumstances as the NFT resides on an immutable blockchain network, and (b) it does not tamper with the 3D model. Finally, transferring the ownership of a watermarked file requires removing the original watermark and inserting a new watermark, whereas our technique naturally supports ownership transfers leveraging smart contract technology. Table I summarizes the advantages of our methodology compared to watermarking.

B. NFTs on Different Blockchain Networks

Although most NFTs are currently based on Ethereum, the blockchain networks and platforms that support NFTs are rapidly expanding (e.g., Bitcoin is planning smart contract support in November 2021). One potential limitation of Ethereum is the transaction cost as gas prices can spike during high demand (e.g., the average gas price on April 2020 was 10 Gwei while two months later went up to 710 Gwei, whereas in January 2022 it varies from 81 up to 218 Gwei).[3] However, many initiatives, including the next major Ethereum upgrade, are focusing on improving Gas costs and scalability. New blockchain networks that already offer support for NFTs and asset management include PRüF [20], Worldwide Asset eXchange (WAX) [21], and Polkadot [22].

Cardano and its *energy-efficient* Proof-of-Stake (PoS) protocol is another open-source blockchain platform. Although Cardano does not officially support NFTs yet, several related projects have launched that leverage the native support of tokens in Cardano. As such, it is expected that Cardano will become one of the leading platforms in the NFT space. Conversely, Bitcoin, Ethereum, and other Proof-of-Work (PoW) protocols have been criticized for consuming excessive electricity to achieve consensus in the network. Thus, Ethereum is now planning to transition from its energy-intensive PoW protocol to PoS, like Cardano. This new era of blockchain protocols is expected to have lower transaction fees, while significantly increasing the number of transactions per second, to enable a broad range of applications. We remark that our solution is independent of the consensus algorithm used (i.e., PoW or PoS) and it can work with any network that supports smart contracts and NFTs.

C. Blockchain Solutions for Supply Chain Management

The evolution of blockchain technology along with its transparency and traceability properties offers an ideal fit for sustainable supply chain management. The authors of [9] analyze four potential barriers for adopting blockchain technologies in the supply chain and highlight the need to represent ownership on the blockchain, especially when different products are traded across multiple actors. Likewise, the authors of [23] focus on how smart contracts can improve the current supply chain in the agriculture sector by tracking shipments and authenticating the origins and the destinations of different products. Nevertheless, these works focus only on transferring goods and how smart contracts can facilitate these processes, failing to address how product ownership is actually represented on a public blockchain, in light of potential counterfeits. Specifically, on top of using smart contracts for supply chain management, we propose a novel methodology for authenticating IPs using frequency domain transformations and NFTs; our approach can also be extended to other file formats beyond STL, such as audio samples.

In the same direction, the authors of [24] discuss the need of integrating blockchain technologies with CAD models to thwart IP theft, and they propose encoding and licensing these models using smart contracts. Their approach, however, can only represent ownership by simply checking equality of the hash of the model's file. Contrary to our frequency-based solution, this hash equality check can easily be bypassed by introducing a minor modification that does not affect the 3D model but changes the binary contents of the CAD file. Thus, a malicious actor can easily produce completely different hashes and claim ownership of a virtually identical model. In this case, our signal processing technique unlocks new possibilities from proving ownership of 3D models, as it relies on shape similarity instead of file contents equality.

VI. Conclusions

The rapid growth of digital manufacturing has attracted new classes of attacks that exploit the lack of existing methods for proving ownership of CAD models and protecting their IP. Existing methods that embed watermarks in the designs remain susceptible to a variety of attacks that allow removing these watermarks and eventually enable counterfeit production. We propose a novel approach for authenticating 3D models using blockchain technologies, namely NFTs and smart contracts, combined with signal processing techniques. Our methodology generates bespoke identifiers for 3D shapes based on their frequency domain representation, which remains resilient to modifications of the corresponding CAD file. Our observation is that frequency-domain similarity comparisons can detect counterfeits even after a series of modifications. To enable public verifiability, we integrate the frequency domain representation of the CAD model into an NFT that lives on the Ethereum blockchain, which enables ownership tracking through the lifetime of a design.

Acknowledgments

This article is based upon work supported by the National Science Foundation (NSF) under Grant CMMI-1931916. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of NSF.

[3] https://etherscan.io/chart/gasprice

Accepted Manuscript
Version of record at: https://doi.org/10.1109/MCE.2022.3164221

REFERENCES

[1] M. Savastano *et al.*, "3-D printing in the spare parts supply chain: an explorative study in the automotive industry," in *Digitally supported innovation*. Springer, 2016, pp. 153–170.

[2] S. C. Joshi and A. A. Sheikh, "3D printing in aerospace and its long-term sustainability," *Virtual and Physical Prototyping*, vol. 10, no. 4, pp. 175–185, 2015.

[3] U. M. Dilberoglu *et al.*, "The role of additive manufacturing in the era of industry 4.0," *Procedia Manufacturing*, vol. 11, pp. 545–554, 2017.

[4] N. Gupta *et al.*, "Additive manufacturing cyber-physical system: Supply chain cybersecurity and risks," *IEEE Access*, vol. 8, pp. 47 322–47 333, 2020.

[5] N. G. Tsoutsos, N. Gupta, and R. Karri, "Cybersecurity road map for digital manufacturing," *Computer*, vol. 53, no. 9, pp. 80–84, 2020.

[6] P. Mahesh *et al.*, "A survey of cybersecurity of digital manufacturing," *Proceedings of the IEEE*, 2020.

[7] S. Voloshynovskiy *et al.*, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," *IEEE Communications Magazine*, vol. 39, no. 8, pp. 118–126, 2001.

[8] M. W. Hatoum *et al.*, "Using deep learning for image watermarking attack," *Signal Processing: Image Communication*, vol. 90, p. 116019, 2021.

[9] S. Saberi *et al.*, "Blockchain technology and its relationships to sustainable supply chain management," *International Journal of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[11] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[12] S. Wang *et al.*, "An overview of smart contract: architecture, applications, and future trends," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 108–113.

[13] I. A. Umoren *et al.*, "Blockchain-Based Energy Trading in Electric-Vehicle-Enabled Microgrids," *IEEE Consumer Electronics Magazine*, vol. 9, no. 6, pp. 66–71, 2020.

[14] D. Mouris *et al.*, "Peak Your Frequency: Advanced Search of 3D CAD Files in the Fourier Domain," *IEEE Access*, vol. 8, pp. 141 481–141 496, 2020.

[15] W. Li *et al.*, "Computer aided design (CAD) model search and retrieval using frequency domain file conversion," *Additive Manufacturing*, vol. 36, p. 101554, 2020.

[16] R. Frank. NFT sales top $2 billion in first quarter. CNBC. [Online]. Available: https://cnb.cx/3mKgNan

[17] W. Entriken *et al.*, "ERC-721 Non-Fungible Token Standard," *Ethereum Foundation*, 2018.

[18] T. Harte and A. Bors, "Watermarking 3D models," in *Proceedings. International Conference on Image Processing*, vol. 3, 2002, pp. 661–664 vol.3.

[19] H. M. Ozaktas and L. Onural, *Three-dimensional television: capture, transmission, display*. Springer, 2007.

[20] PRüF. Prüf a blockchain based asset tokenization ecosystem. [Online]. Available: https://pruf.io/assets/files/PRuF-whitepaper1.pdf

[21] W. Quigley *et al.* Worldwide asset exchange protocol white paper. [Online]. Available: https://github.com/worldwide-asset-exchange/whitepaper

[22] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White Paper*, 2016.

[23] R. Casado-Vara *et al.*, "How blockchain improves the supply chain: case study alimentary supply chain," *Procedia computer science*, vol. 134, pp. 393–398, 2018.

[24] F. Engelmann *et al.*, "Intellectual property protection and licensing of 3D print with blockchain technology," *Transdisciplinary Engineering Methods for Social Innovation of Industry*, vol. 4, pp. 103–112, 2018.

Dimitris Mouris is a Ph.D. student with the Electrical and Computer Engineering department at the University of Delaware. He is a Graduate Student Member of IEEE. Contact email: jimouris@udel.edu.

Nektarios G. Tsoutsos is an assistant professor of Electrical and Computer Engineering at the University of Delaware. He is a Member of IEEE. Contact email: tsoutsos@udel.edu.