TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
46/2020

Smart Cyber-Physical System for Personal Manufacturing

ANTON VEDEŠIN

TAL TECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

The dissertation was accepted for the defense of the degree of Doctor of Philosophy (Computer Science) on the 27th of October 2020

Supervisor: Innar Liiv, Associate Professor
Department of Software Science,
Tallinn University of Technology,
Tallinn, Estonia

Co-supervisor: Dirk Draheim, Professor
Department of Software Science,
Tallinn University of Technology,
Tallinn, Estonia

Opponents: Dieter Kranzlmüller, Professor
Ludwig-Maximilians University of Munich,
München, Germany

Josef Küng, Professor
Johannes Kepler University Linz
Linz, Austria

Defence of the thesis: 7 December 2020, Tallinn

Declaration:
Hereby, I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Anton Vedešin _____
signature

ISSN 2585-6898 (publication)
ISBN 978-9949-83-627-7 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9949-83-628-4 (PDF)
Printed by Auratrükk

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
46/2020

Tark küberfüüsikaline süsteem personaalseks tootmiseks

ANTON VEDEŠIN

TAL TECH
KIRJASTUS

Contents

List of Publications

The present Ph.D. thesis is based on the following publications that are referred to in the text by Roman numbers.

I A. Vedeshin, J. M. U. Dogru, I. Liiv, D. Draheim, and S. Ben Yahia. A digital ecosystem for personal manufacturing: An architecture for cloud-based distributed manufacturing operating systems. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, MEDES '19, page 224–228, New York, NY, USA, 2019. Association for Computing Machinery

II A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. *IEEE Access*, 2019

III A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. Smart cyber-physical system for pattern recognition of illegal 3d designs in 3d printing. In *Communications in Computer and Information Science*, pages 74–85. Springer International Publishing, 2020

IV P.-M. Sepp, A. Vedeshin, and P. Dutt. Intellectual property protection of 3d printing using secured streaming. In *The Future of Law and eTechnologies*, pages 81–109. Springer, 2016

List of Patent Applications

The present Ph.D. thesis is supported by the following patent applications that are referred to in the text by Roman numbers.

V K. Isbjornssund and A. Vedeshin. Method and system for enforcing 3d restricted rights in a rapid manufacturing and prototyping environment, Feb. 27 2014. US Patent App. 13/973,816

VI K. Isbjörnssund and A. Vedeshin. Secure streaming method in a numerically controlled manufacturing system, and a secure numerically controlled manufacturing system, Dec. 3 2015. US Patent App. 14/761,588

Author's Contributions to the Publications

I In **I**, I was the main author, conducted the experiments and simulations, wrote numerous parts of the software, and analyzed the results, prepared the figures, and wrote the manuscript.

II In **II**, I was the main author, conducted the experiments and simulations, wrote the software implementation, analyzed the results, prepared the figures, and wrote the manuscript.

III In **III**, I was the main author, wrote several parts of the simulation program, conducted the experiments and simulations, analyzed the results, prepared the figures, and wrote the manuscript.

IV In **IV**, I was the main technical author, conducted the experiments and simulations, carried out the analysis of the results, prepared the figures, and wrote sections 1, 2, 3, 5 and 6 of the manuscript.

Author's Contributions to Patent Applications*

V In **V**, I was the main technical author, analyzed prior art, conducted the experiments and simulations, analyzed the results, prepared the original figures later used by the patent attorney to convert to black-and-white patent pictures, created technical workarounds for close and similar patents, and wrote several parts of the manuscript.

VI In **VI**, I was the main technical author, analyzed prior art, conducted the experiments and simulations, analyzed the results, prepared the original figures later used by the patent attorney to convert to black-and-white patent pictures, created technical workarounds for close and similar patents, and wrote several parts of the manuscript.

*As of 2020, those patent applications have been cited by 83 other patent applications, patents, and scientific articles, including these well-known companies (number of citations is shown in parenthesis if more than one citation): Amazon (12), Hewlett-Packard (4), Accenture (3), IBM (3), Caterpillar (2), Walmart (2), Adobe, Airbus, Aurora Labs, Canon, Capital One, Centurylink, Disney, eBay, Eos, GE, Hitachi, Intel, Jabil, Mitsubishi, Qualcomm, Siemens, Sony, Western Union.

Abbreviations

CAD	computer-aided design
CAE	computer-aided engineering
CAM	computer-aided manufacturing
CAPP	computer-aided process planning
CAQ	computer-aided quality assurance
PPC	production planning and control
ERP	enterprise resource planning
API	application programming interface
AM	automated manufacturing
IP	intellectual property
NDN	named data network
FFF	fused filament fabrication
DLP	digital light processing
SLA	Stereolithography
SLS	selective laser sintering
SLM	selective laser melting
DED	directed energy deposition

Terms and Definitions

STL	an STL file describes a raw, unstructured triangulated surface by the unit normal and vertices (ordered by the right-hand rule) of the triangles using a three-dimensional Cartesian coordinate system [11]
Personal Manufacturing	using own portable machinery to manufacture products at the time and point of need [83]

Introduction

This thesis is structured in the following way. In section 1, we establish the motivation and provide the problem statement behind this work. In section 2, we state the contributions of the thesis from three different angles: ecosystem, secured infrastructure and application layer with illegal object detection and intellectual property protection. In section 3, we describe the methodological approach we used. In section 4, we discuss related work for publications **I**, **II** and **III**. In section 5, we explain how publications **I**, **II**, **III**, **IV**, **V** and **VI** contribute to this thesis, creating a synergic effect and (as a whole) contributing to a larger phenomenon—the development of a Smart Cyber-Physical System for Personal Manufacturing. In section 6 we conclude and discuss directions for further research.

1 Motivation and Problem Statement

There are four main problematic areas of automated manufacturing (AM):

- **Fragmentation and interconnectivity**. 3D printers today are the same stage as computers were 60 years ago, and mobile phones 15 years ago. Historically, each hardware manufacturer created an operating system to operate its hardware [3, 4]. Only with the development of UNIX [103] for mainframes, MS-DOS [8] for personal computers, and Android [1] for mobile phones were these areas democratized. Anyone could create an application that would run on different hardware managed by a standard operating system. With UNIX operating systems, mainframes from different hardware manufacturers could run the same programs. UNIX advanced through universities; it replaced numerous operating systems at universities removing fragmentation [3]. Android changed the smartphone landscape and eliminated the fragmentation of numerous operating systems for mobile phones 12 years ago [35]. There are a few initiatives to standardize 3D printing file formats (e.g., STEP, AMF, 3MF) [113]. However, there has so far been no overall standardized approach to control the end-to-end process of 3D printing.

- **The social change: the era of personal manufacturing**. Every human being is inherently a manufacturer. Today, society has reached an inflection point in AM—*personal manufacturing* [64, 83, 102, 105, 106]. Society has accumulated knowledge and a sufficient level of technological development in manufacturing, tools, and their implementation in the form of cloud manufacturing. Now, home users, small, medium, and Fortune 2000 enterprises use devices such as 3D printers, computer numerical control (CNC) mills, laserjets, and robotics to manufacture products locally, at the point and time of need. Ease of use is important—and now, a physical part can be produced with a single click of the computer mouse [15].

- **IP infringement and manufacture of illegal and regulated objects**. Today, manufacturing files contain all of the intellectual property (IP) needed to produce a production-grade physical part. This was never the case even 20 years ago [116]. This vital IP needs protection to secure the IP of companies manufacturing products. Another important aspect is protecting society from the unregulated manufacture of harmful objects like firearms, weapons, and ammunition [117].

- **Large CO2 footprint**. Manufacturing has its share of CO_2 footprint, and if manufacturing facilities were connected and virtually accessible through the cloud, we would reduce manufacturing CO_2 footprint portion. Our solution supports a shared

economy and distributed manufacturing, allowing us to reuse securely existing manufacturing facilities [115]. It is part of our vision that the CO_2 footprint of the proposed solution is a lot less. From the energy waste and CO_2 footprint perspective, it is much better to manufacture a part at home or in close proximity using locally available materials [70, 121] versus shipping it around the globe with logistics companies and airplanes. Even though frequently securing the 3D designs can potentially increase the CO_2 footprint, the cost of loss of the important digital content prepared to be used in advanced manufacturing machines has much higher importance than the energy consumed. For example, blockchains and Bitcoin particularly use a proof-of-work consensus algorithm [46], which also consumes a lot of energy and creates a considerable CO_2 footprint. However, this allows us to solve the problem of trust and keep the system functioning; moreover, compared to paper or metal money printed in every single country, thousands of banks keeping monetary balances in their databases and data centers could be comparable with the CO2 footprint of cryptocurrencies. Another example, COVID-19, showed that people were flying too much, driving too much. Many people can create a smaller CO_2 footprint by working and making video calls from home, versus flying to business meetings or driving every day to the office.

The authors believe that the future of manufacturing is one secure Smart Cyber-Physical System for Personal Manufacturing, which covers the whole end-to-end product creation process from idea to a physical object is presented in Figure 1.

Figure 1 – Smart Cyber-Physical System for Personal Manufacturing, which covers the whole end-to-end product creation process from idea to a physical object.

2 Contribution of the Thesis

We contribute to the "democratization" (in the sense of bringing the production to the end-user [115]) of *personal manufacturing* [83] from multiple different angles: ecosystem, security, IP protection, and safety, as follows:

- Addresses evolving critical problems of personal manufacturing: democratization, security, IP protection, and safety;
- Proposes the next step in the evolution of user participation in manufacturing—*personal manufacturing*;
- Proposes and develops a unique, secure, and self-sufficient software ecosystem for *personal manufacturing*. Such a system would support all of the components necessary to move from an idea to a physical object in one click;
- Introduces a novel, cloud-based software ecosystem capable of sustaining a massive communication load of command, control, and telemetry data coming to and from millions of manufacturing machines and users. Our solution allows users to create and deploy their applications in a cloud operating system;

- Contribute a secure and dependable infrastructure and architecture for the new economy of the future - *personal manufacturing*;
- Proposes an approach that leverages the computational process's physical limitations into a defense strategy that makes distributed file storage and transfer highly secure, a novel, key-less, byte-less encryption method, ready for application to AM;
- Offers a threat model and security analysis of the proposed approach;
- Cloud-based manufacturing operating system to address an evolving critical problem of IP protection and manufacture of illegal physical objects;
- Introduces a novel cloud-based manufacturing operating system architecture to protect IP and detect illegal physical objects using pattern recognition.

Taken together, these contributions commit to:

- democratizing automated manufacturing (AM);
- reducing the prototyping and manufacturing latency of new and existing products on the market;
- securing communication with AM machines to protect them from sabotage when producing mission-critical parts;
- protecting society from the threat of manufacture and distribution of illegal objects.

3 Methodological Approach

The main body of this thesis is composed of four original peer-reviewed articles: **II** is an article in journal, **I** and **III** are classified as articles in conference proceedings, and article **IV** is a book chapter. There are additional publications which are patent applications **V** and **VI** which strongly support authors' point of view. Each article is based on a separate empirical study, however altogether they form a theoretical and practical framework integrated into Section 5 of this thesis. The theoretical and practical framework is based on a synthesis of existing research on different aspects of smart cyber-physical systems for automated manufacturing, such as: a) an ecosystem layer - a fundamental orchestration layer enabling all other parts of the ecosystem to work, the details are covered in publication **I**; b) a secure storage and communication layer interconnecting data security critical parts of the ecosystem, the details are covered in publication **II**; c) a practical application layer, e.g. pattern recognition of illegal designs and intellectual property infringements, the details are covered in publication **III**.

According to the Methodology framework for the design of digital ecosystems [51] a digital ecosystem should have several key factors associated with the digital ecosystem design, e.g., roles of different digital components within a system, organization, and collaboration of the digital components, their individual design along with intelligence and security within the system. In this thesis, we have covered all the key aspects. According to the model-based design methodology for complex, distributed across networks cyber-physical systems [60], the cyber-physical system described in this thesis was evolving in line with the ten steps of the methodology [60], namely: 1) we stated the problem of software fragmentation and inter-connectivity for automated manufacturing; 2) we studied and modeled physical processes of multiple types of manufacturing machines, such as FFF, DLP, SLA, SLS, SLM, DED; 3) we characterized the problem by isolating fixed parameters,

adjustable parameters and variables to be controlled, identified quantities that characterize physical processes, such as material melting temperatures, deposition speeds, movement speeds, accelerations, slicing parameters, and many more; 4) we derived control algorithms by determining under which conditions physical processes are controllable; we used the problem characterization to specify requirements on the heating and cooling times, network latencies, 3D printer firmware buffer sizes, specifics of electronics, and mechanics of different manufacturing machine types; 5) we have selected different models of computation for different types of manufacturing machines, as each distinct type has its own set of allowable instructions used in a computation along with rules that govern interaction, communication, and control flow; 6) the hardware was specified by the 3D printers and other manufacturing machines manufacturers; thus, we adjusted computational models for each hardware component and manufacturer; this is one of the fundamental complexity we faced during the development of a cyber-physical system; 7) we have created our simulation tools for different types of automated manufacturing machines helping us to verify and visualize the correctness of computational models; 8) the AM machine manufacturers constructed the devices, however frequently we needed to adjust the hardware and their hardware decisions to more easily make it work with the existing computational models and control algorithms; 9) we developed software according to the specifics of computational models and control algorithms; moreover, we generalized and unified software components; 10) we verified, validated, and tested each component of the system and subsystems independently and combined, sometimes we need to use 100+ the same devices to perform tests to get adequate results. We followed the process multiple times in iterations with an average length of 1 year. This way, we achieved stable, predictable results, every time improving all aspects of the cyber-physical system.

The empirical evidence of the thesis derives mainly from quantitative analysis. The methodological approach is the following: a) in **I**, we used quantitative methods that emphasize objective measurements and the statistical, mathematical, and numerical analysis of collected user data over multiple years; b) in **II**, we used both qualitative and quantitative methods to perform threat modeling, security analysis, and performance evaluation; c) in **III**, we applied quantitative methods to a dataset of 3D models to perform classification of these models into loosely coupled classes and prediction of potentially harmful objects by using statistical and mathematical analysis; d) in **IV**, we did a theoretical study, and we did not conduct statistical and mathematical analysis.

4 Related Work

We have analyzed the most influential works on the topic from multiple angles. We have structured the discussion of related work by first describing related work for publication **I**, then publication **II**, and so on.

For publication I, we analyzed the most influential and highly cited works published on cloud manufacturing [101], social aspects of advanced manufacturing, and interoperability.

In their work "Advanced manufacturing systems: socialization characteristics and trends", F. Tao et al. [105] analyze the degree and scope of resource sharing since the 1960s. Then, they describe four phases in the evolution of manufacturing resource sharing. Moreover, they describe the degree of user participation in manufacturing a product: Buy; Buy and choose [102]; Buy, choose, and design; and Full customization. In our work, we seek to contribute the next logical step of the evolution of consumer participation in manufacturing by creating a new model—*personal manufacturing*.

Tao et al. [106] define and compare cloud manufacturing to cloud computing and out-

line the key advantages of the former. One of the most important is the generation of new types of business models and ways to deliver products [127]—for example, RESA [91] and MyStemKits [84]. The latter utilizes our solution's API [87].

In their respective works, Tao et al. [105], Ray et al. [90], Tibaut et al. [108], Wang et al. [120], Panetto et al. [88], and Figay et al. [44] name the interoperability of manufacturing systems and components as one of the most compelling challenges in the evolution of cloud manufacturing resource sharing. Interoperability requirements affect the architecture of manufacturing cloud operating systems. In our work, we utilize protocols that integrate new manufacturing machines as they become available.

Kalpakjian et al. [62] describe a manufacturing ecosystem that employs computers to manage and control the entire manufacturing process through CAD, CAE, CAM, CAPP, CAQ, PPC, ERP (see Abbreviations section) and the ability to create AM processes utilizing all or some of the steps mentioned in Figure 1. The degree of user participation and interaction with manufacturing enterprises is increasing, and "network topologies and value creation increasingly reflect the trend of universal participation and social manufacturing".

The Kozmetsky effect [64] occurs whenever the specific knowledge and technology involved in a specific industry reaches a saturation state. The strong interaction of knowledge, technology, tools, and implementation stimulates creativity and innovation. Latency is diminished among all four components (knowledge, technology, tools, implementation) and becomes zero. New ideas and technologies arise under these conditions, new product prototypes appear, and new market segments emerge. This is the beginning of a strategic inflection period when all the new things are put on trial that creates divergence, confusion, and chaotic situations [64]. This phenomenon is currently noticeable in the field of AM. Society has accumulated enough knowledge and a sufficient level of technological development in manufacturing, tools, and their implementation in the form of cloud manufacturing to reach an inflection point of this effect. The era of *personal manufacturing* has just begun. Now, home users, small, medium, and Fortune 2000 enterprises use devices such as 3D printers, CNC mills, laserjets, and robotics to manufacture products locally, at the point and time of need. Our contribution to the field is 3DPrinterOS—a digital ecosystem for personal manufacturing.

For Publication II, we start with some general considerations of cloud security, and then go more deeply into specific solutions, like point-to-point and point-to-multipoint secured communication, cloud secured storage, digital rights management (DRM), video streaming, and 3D streaming.

Over time, there has been a trend in AM to move as much calculation to the cloud as possible due to the low cost of cloud computing power. Initially, slicing for 3D printers was performed on the workstation built into a 3D printer (e.g., [10, 12]). Then, slicing software moved to engineers' workstations [6]. Now, slicing has moved to the cloud [15], with machine code streamed to the AM machine. The next important step is to stream stepper motor pulses from the cloud directly to the AM machine; thus, the firmware moves to the cloud. As with software and faster computing, this move improves hardware operation, with incredible increases in quality and speed. For instance, Okwudire et al. [86] sent low-level stepper motor commands from a server to simplified firmware, which interpreted simple commands and proxied them to the stepper motor drivers and measured an increase in printing quality and speed.

AM machines should have a thin client built-in, not a workstation [10, 12]. This thin client will interpret commands and send current status and metrics to the cloud. Moving path planning out of the firmware to a nearby computer increases manufacturing speed

and quality. This was achieved by the team of researchers behind the Klipper project [5]. Their table of step benchmarks [5] shows that the same hardware can be 10× more efficient with the right software and more computing power. To achieve such improvements, we will ultimately stream encoded physical signal commands from the cloud to AM machines. The method proposed in the publication **II** is ready for these types of applications.

In [31], Brunette et al. provide a comprehensive analysis of possible cloud security issues and suggest mitigation strategies. They present a robust approach to assess existing cloud applications and provide a requirement base for the design of secure cloud solutions. That work provides notable recommendations. However, from our perspective, the next level—an integral solution—is necessary. For the sake of an ultimate security solution for cloud storage and file transfer, we need a change in philosophy and a new paradigm—*live matrix*—which we describe in Publication **II**.

We examined related research on peer-to-peer, point-to-point, and point-to-multipoint communication. First, most such solutions tend to use the Open Systems Interconnection (OSI) [66] model's lower layers, mostly layer 3, the network layer. This increases the communication speed and throughput. Simultaneously, it makes most of the protocols proprietary and exotic, making a wider implementation for AM machines difficult. In contrast, the solution we propose in this paper is the network layer and is protocol agnostic, as the only information that is transferred is cryptographic hashes. Our solution would benefit from using a lower layer of the OSI model, and streaming hashes over a lower level of the OSI model is a topic worthy of future research and experiments. The primary efforts in the literature are focused on the resolution of peers and re-routing if a peer is disconnected. These mechanisms can complement the solution described in this paper. Many point-to-point and point-to-multipoint communication security approaches employ basic private/public key encryption, which does not prevent intellectual property exposure.

Mastorakis et al. [77, 78] discuss peer-to-peer file sharing application designs and implementations that run on top of Named Data Networking (NDN). The security aspect is in the nature of the NDN architecture; however, this suggests the cryptographic signing of every packet in the network. NDN uses a distribution of data encryption keys as its encrypted data. Because it implements security at the protocol level, NDN offers protection against negligence, in contrast to TCP/IP, where applications are responsible for security. Although NDN is considered the future of the Internet [125], it is still a work in progress and not yet ready for full production-grade implementation.

In their cryptographic protocol [58], Jaatun et al. present an approach that is similar to ours. They segment files among a redundant array of independent net storage containers in the computing cloud. The main thrust of their solution is the distribution of data across different cloud providers. Thus, the individual data deposits do not expose enough information about the owner and the file to make them vulnerable. The data must be reassembled to return the file to the user. In our approach, we similarly distribute file parts to many machines in the cloud; however, we do not set a specific constraint on the form and number of cloud providers; our approach can utilize physical computing machines, virtual machines, Docker containers from one or several providers, or other solutions.

Miller et al. [81] propose several robust security schemes for distributed file systems. They use the segmentation of files into file blocks and file block encryption with asymmetric keys. We split a file into segments and encrypt each segment with its key in a manner similar to [81]. However, we go beyond this and propose continuous re-encryption of file segments, with continually changing keys. Moreover, we may continuously re-encrypt the symmetric keys that data segments are encrypted with. In our approach, re-encryption happens continuously on all cloud nodes at a preset file, computational, or cost limit.

In [18], Giuseppe et al. describe improved proxy re-encryption schemes for keys and apply them to secured distributed storage. We apply a similar approach in our solution, but to file segments as well as keys. Furthermore, we re-encrypt continuously, regardless of reads and writes to storage. Cloud computing infrastructure prices drop each year, making such a re-encryption approach feasible for use with millions of files.

Many practical DRM-like approaches are widely used in cloud storage and transfer. These include effective cryptographic file system (ECFS) [92], and other methods mentioned in the same paper. In DRM, a file is usually encrypted using a symmetric or asymmetric key or a key combination before it is stored or transferred. Symmetric encryption is not a solution here as follows. The problem is not to protect data only against a man-in-the-middle attack but against the user himself/herself. In order to access the file, the data consumer needs the key. When an attacker obtains the key by, for example, buying the protected content once, brute force, or social engineering attacks, the file can be used or redistributed infinitely. Thus, the user must produce the part a limited number of times, e.g., one time. This is a wicked problem that needs a complex solution. The use of symmetric and asymmetric encryption schemes assumes that secret keys will never be exposed or calculated. This is also a standard solution for DRM-like methods to protect digital content; however, these have failed over time, especially when protecting against the end-user device. DRM methods are usually lightweight and can be functional without any need for intensive cloud computing power. From our perspective, DRM methods are too vulnerable by their nature. Our goal is to make information disposable, time-dependent, constrained to reproduce the digital content only once at the end-user device side, and at the same to make it extremely hard to get the original content for mass reproduction without the owner license.

Numerous existing streaming approaches [32, 37, 72, 74, 122] work efficiently and consistently for video and music. Even though some of the protocols have consistency checks, they are not expected to deliver every single byte; insignificant data loss or delay caused by network problems is expected. However, such losses could be an issue for sensitive data, like CAD designs. For example, in the case of streaming designs to automatic manufacturing machines such as 3D printers or CNC mills, data transfer should be consistent and lossless: loss of a single byte while streaming is unacceptable, as this can lead to an AM machine malfunction or a defective product. Simultaneously, the streaming should be highly secured, which is not usually required for media streaming protocols. In publication **II**, we show how to securely stream encrypted file segments directly from a highly secure distributed file storage.

In [71], Lin et al. describe a method to encode 3D models into a JPEG stream to transfer 3D designs. However, the solution is not comprehensive and has definite limitations.

In prior research [100], we theoretically described *live matrix* as a paradigm applied to secured 3D content delivery. Our prior work is purely theoretical, lacking technical details and a real implementation of the method. This paper's [100] contribution extends the initial idea with the necessary details for implementation and to technically broaden it to any type of secure file storage and transfer. Furthermore, we describe a threat model and conduct a thorough security analysis and eliminate the transcoding of files for streaming introduced in prior work [56, 57].

We previously explained in detail the necessity for the secured streaming of 3D files and discussed methods to enforce 3D file copyrights [56, 57]. In that approach, we targeted a small niche case to secure 3D design transfer to 3D printers. That solution is machine code-centric and lacks a tight coupling with the secured storage. Furthermore, it is vulnerable at the point of extracting a 3D design from the storage and re-encoding it for

streaming. In the publication **II**, we propose a much more secure and consistent end-to-end method to store and stream files—regardless of file type—and without the need to re-encode the file for streaming.

We compare our solution with cryptosystems with symmetric and asymmetric encryption [109]. As described in details in publication **II**, our solution does not have a decryption function in terms of symmetric and asymmetric encryption, as there is no key as such in terms of that equation, and actual static bytes are not transferred in its terms.

Our solution depends on time synchronization, though its dependency on time could be entirely removed if we synchronize against other sources. Future accessible synchronization methods might include natural phenomena like geomagnetic micropulsations [45, 59], seismic activity, gravity, blockchain block number, shared prior quantum entanglement [61, 124], and others.

Computation overhead and CO2 footprint matters. Our solution has a bigger positive overhead compared to well-known stream ciphers [16, 23, 26, 41, 42, 48, 52], block ciphers [27, 39, 50, 99] and plain text in terms of computation and bandwidth.

Commonly, in stream ciphers [94] the ciphertext length has an insignificant positive overhead compared to plain text. In block ciphers [73], padding is frequently added to plain text to make it equal to the block size, increasing the bandwidth overhead. Block ciphers also have computational overhead to encrypt each block compared to plain text. Our solution has a parametric trade-off between computational complexity and security. Our approach's computational overhead is lower than that of block ciphers, and depending on the stream cipher algorithm, it can be smaller than stream ciphers. Hash function calculation is less expensive than AES and DES [110, 123]. Another possibility is to scale hash calculation using a GPU and an application-specific integrated circuit (ASIC) implementation of hash functions [36, 38]. However, block and stream encryption are difficult to implement using a GPU- and ASIC-based approach.

The philosophy behind our solution is to set an attacker against a computing cloud and leverage the physical limitations of the computational process [65] as security controls. Similar to proof-of-work blockchain consensus algorithms [46], we parameterize the solution based on the amount of available infrastructure.

There is physical latency at all levels of hardware and software during computational processes. In order to reduce latency, computer L1 and L2 cache memory are located very close to the processor [49]. The more distant some resource is from the processor, the higher the latency. For example, a network interface is usually the main bottleneck for distributed systems [126]. The operating system limit of open ports and I/O descriptors in Linux can be a bottleneck [30]. Our approach is to use these limitations and bottlenecks and turn them into a defense strategy.

Our threat modeling and security analysis are based on several well-defined threat frameworks from Behl et al. [20–22] and Saripalli et al. [95]. The latter provides the list of "Threat events compromising cloud security" [95], which our distributed storage and transfer solution are intended to address.

In our previous research [56, 57, 100], we assumed that data—once taken from some kind of secured storage—are decrypted and then encrypted with a different method for delivery to the data consumer. Then, the distribution node can also be a point of attack. In that case, the attacker can obtain a file or a stream on the server during re-encryption between storage and streaming. However, in the current approach, there is no need for transcoding. In our solution, we use elliptic curve type secp256k1 [28] for public/private key generation. For hashing, we use the Keccak-256 [25] hash function.

For Publication III we discuss related work and offer an overview of existing methods

to protect AM-related IP.

In their work, Hou et al. [54] cover traditional ways to secure 3D files. The authors describe solutions ranging from the application of DRM to embedding visual shapes into 3D printed models' internal structure. In the latter approach, scanning the item's internal structures allows determining whether it is an original part or a copy. Their work does not cover the whole AM workflow and focuses on protection methods that can help after manufacturing, such as watermarking and tagging an object with RFID chips. Although these approaches can help users detect whether the part is original, it does not protect parts from being copied or prevent others from manufacturing an illegal part. Moreover, it is hard to reverse engineer a 3D-printed part by scanning it due to internal structures. It is insufficient to obtain the shape of the object; reproducing it requires knowledge of the 3D printer's toolhead movements, speed, and the temperature used at that exact path. All of these factors are important and affect the final physical properties of the manufactured object. The protection of this IP is essential and must take place before manufacturing.

Research from Nein-Hsien et al. [71] describes a method to encode 3D models into a JPEG stream for the transfer of 3D designs. This is not a comprehensive solution for IP issues and has definite limitations. Their solution does not handle end-to-end AM workflow, nor does it prevent the manufacture of specific 3D designs.

In our prior research [100], we only theoretically touched upon the protection of IP rights in 3D printing. We have described a paradigm applied to secure 3D content delivery called *live matrix*. This prior work is purely theoretical and lacks technical details. The current paper is the first to explain its technical application to IP protection and detection of illegal 3D parts. Thus, this paper's contribution is to extend the initial idea with details of its implementation and technically broaden its application to the detection of illegal 3D parts before such parts are manufactured.

In previous work [56, 57, 116], we have emphasized the necessity to enforce 3D files' copyrights through secured content delivery—3D file streaming. Our previous work has targeted a very niche case to secure 3D designs' data at rest and while transferring from a server to a 3D printer. Previous solutions [56, 57] are technically dense and face multiple drawbacks. Moreover, they do not offer IP protection at any other stage, nor do they detect illegal 3D parts or prevent their manufacture [116]. In this paper, we contribute a reliable and fast way to detect IP infringement and extend the same approach to detecting illegal physical objects and the prevention of their manufacture.

The use of blockchain technology has been widely suggested for IP protection for 3D designs. For instance, Mattingly et al. [79] have described a method for 3D printing with blockchain controls. Their work does not precisely describe how the copyright is given and what happens with the file. The idea of just closing the block with the list of transactions is part of blockchain's nature; however, it is not clear how this method grants access to IP. The solution they propose is a log file containing records of what has happened in the past, regardless of whether or not the IP owner authorized the transaction. The details of implementation are not provided.

Similarly, Holland et al. [53] propose a blockchain-based approach for copyright protection in additive manufacturing. They utilize the blockchain to eliminate third parties; instead, the blockchain acts as a trusted third party or notary to govern transaction data exchange. In their work, 3D design files are not stored in the blockchain, and the blockchain is used only to grant licenses and control the number of prints allowed. However, their method lacks a process for the 3D printer to report to the blockchain. The 3D printer should send a transaction to update the number of times the file was printed and reduce the number of licenses available.

Kennedy et al. [63] present a method that adds a distinctive nanomaterial chemical signature to the parts and registers it in the blockchain. Their method helps to check after the file was manufactured, whether it is an original file or a counterfeit. Their solution does not defend the file nor detect and prevent an illegal 3D file from being manufactured. In their work, the blockchain is a shared resource between the manufacturer of a 3D part and a part-receiving party; both have access to the blockchain instance and can see the transactions made by the manufacturer. From the description of Kennedy et al., it is not clear what would stop the user from updating blockchain data without receiving the part. Confirmation of receiving (or not) the part does not provide much security. Their solution mostly solves audit and integrity problems and works best for checking if a part is genuine after manufacture or during use.

Many other researchers [17, 19, 47] have described how users upload a file to the cloud and how such designs can be protected using DRM; however, none have presented a viable solution to protect the copyright of 3D files in the long term. The works cited here do not describe how to protect files from being exposed to third parties, nor do they offer the potential to detect and prevent the manufacture of illegal parts. Our solution proposes a real solution to enforce copyright protection, detect illegal parts, and prevent the manufacture of inappropriate or unauthorized parts.

While it is essential to allow users and manufacturers to determine if there are any restrictions on reproducing a specific 3D object, ideally, there should also be a mechanism to prevent the unauthorized reproduction of particular 3D objects, especially when such designs represent an illegal or dangerous part, like a firearm. The 3D file represents the 3D object but does not necessarily offer any way to prevent its unauthorized use. Any means of authorization must be integrated with the manufacturing device itself—for instance, by requiring that the manufacturing device seek authorization from the rights holder or confirmation that there are no restrictions on using that file before each manufacturing job. However, the use of our cloud manufacturing operating system and ecosystem [115], detection of, and protection from illegal parts manufacturing can be performed in the cloud. In the case of our proposed method, there is no need for significant modifications on the AM machine side to support a safe and legal method of manufacture.

5 Smart Cyber-Physical System for Personal Manufacturing

Taken together, publications **I**– **VI** form a Smart Cyber-Physical System for Personal Manufacturing depicted in Figure 2. The bottom, ecosystem layer described in detail in publication **I**, is a fundamental orchestration layer enabling the work of the whole ecosystem: end-users, vendors, external actors, niche players, and software components. The software components reside in or communicate with the orchestrator's platform component, described in detail in publication **I**. The second layer is a secure communication layer, described in detail in publication **II**, which provides a secure data infrastructure for secured communication between cloud to cloud node and cloud to a manufacturing machine. The initial idea of the second layer is mentioned in publications **IV**– **VI**. The third layer, pattern recognition of illegal 3D designs in 3D printing, described in publication **III**, is the application layer. These three main building blocks are connected and form the self-sufficient Smart Cyber-Physical System for Personal Manufacturing outlined in Fig. 2.

We define Smart Cyber-Physical System for Personal Manufacturing as follows: interconnected hardware and software components that allow a user to fully manufacture an end product at point and time of need. This Smart Cyber-Physical System for Personal Manufacturing as described in publication **I** and depicted in Figure 3 is interconnecting and manages different types of automated manufacturing machines, 3D printer firmware,

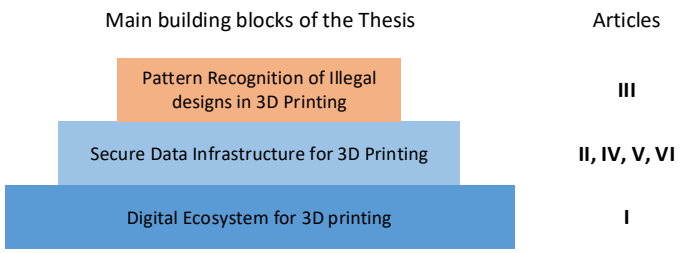


*Figure 2 – Publications **I**– **VI** form a Smart Cyber-Physical System for Personal Manufacturing.*

CAD software, CAD simulation, and manufacturability assessment software, manufacturing preparation software, slicing and toolpath generation, finding the nearest facility with the required machines, manufacturing planning and remotely controlling and driving the machines, manufacturing machines telemetry, real-time quality control, consistency and repeatability adjustment applications, machine learning applications for instant quoting and prediction of the manufacturing and postprocessing time, postprocessing remote process management, final product assembly, shipment to an end customer or a different facility, manufacturing material resource management, user access management. The details of secured connection as described in publications **II**, **IV** – **VI** cover the communication between the printer or a CNC mill and a cloud platform. The details of pattern recognition of illegal designs in 3D printing and copyright protection, as described in **III**, cover the copyright protection and firearm components of the platform.

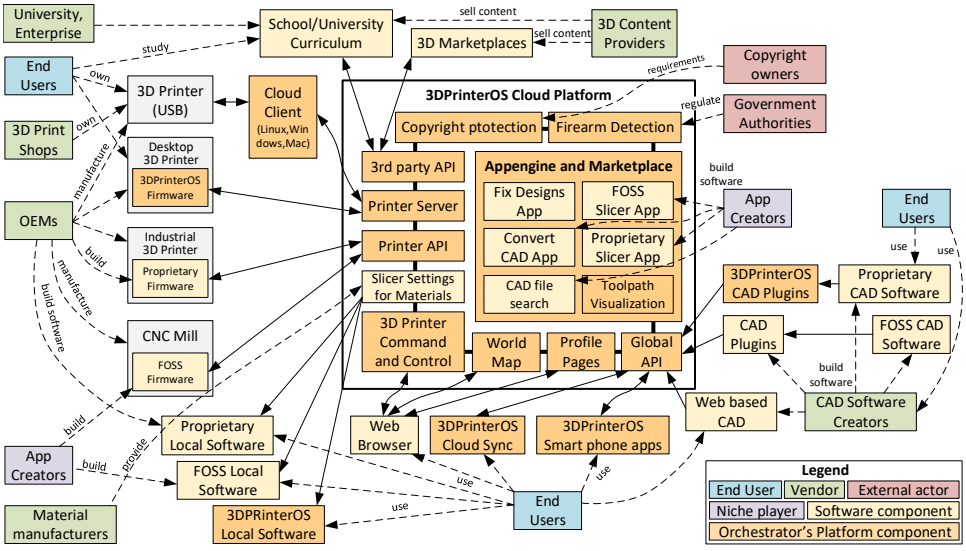


*Figure 3 – Smart Cyber-Physical System for Personal Manufacturing landscape, which is described in details in Publication **I**.*

5.1 Digital Ecosystem for 3D Printing

Publication I reports a case of a digital ecosystem for personal manufacturing developed by the authors over the last seven years. This ecosystem is currently either used or in trials among 111 of the Global F2000 enterprises. In this context, the authors discuss the moti-

vation for the creation of the cloud-based manufacturing operating system, propose the next step of the evolution process for user participation, introduce a novel cloud-based manufacturing operating system architecture, and propose a unique end-user experience for the automated manufacturing flow shown in Figure 1.

Work related to the Publication **I** reviews social aspects of manufacturing [101, 102, 105], cloud manufacturing [84, 91, 106, 127], and the interoperability of manufacturing systems [44, 88, 90, 105, 108, 120].

In **I**, we describe the architecture with cloud kernel, libraries, application framework, basic functionality and 3D app layers. The latter, through its encapsulation into Linux containers [80], enables a high level of security [34, 76]. We define roles, artifacts and relations in 3DPrinterOS ecosystem [75] and classify accordingly [29] into vendors, end-users, external actors, niche players, software components and orchestrator's platform components.

Publication **I** describes two types of experiments: long and short term. During a long experiment that lasted five years, more than $125,000$ end users have produced over five million CAD designs and machine codes and manufactured more than a 1.5 million parts on 40 thousand 3D printers and other AM machines in 100 countries. The extensively tested software components are licensed to vendors (3D printer manufacturers and companies) including Google, Microsoft [40], John Deere, Bosch [82], Dremel, U.S. NAVY, NASA, Kodak [97], MilleBot [98], Robo3D [96], Loop3D [43], Imprinta and others, and distributed to their end-users. The first short-term experiment (74 participants) showed that, on average, it takes five times less effort (compared to CURA [111]) to produce a physical part using the proposed software ecosystem. The second experiment (12 elementary school students were involved with the permission from their parents) showed that all participants could not perform a 3D print on an Ultimaker 2 [112] using 3D printing software such as CURA [111], Octoprint [89], Repetier-Host [9]. In contrast, 11 out of 12 students were able to prepare and perform a 3D print using 3DPrinterOS [93].

5.2 Secure Data Infrastructure for 3D Printing

Publication **II** proposes a secure file delivery solution for highly secure distributed file storage and transfer used for manufacturing devices such as 3D printers and CNC.

In **II**, **IV**, **V**, and **VI**, the authors describe why security is extremely important for the domain of personal manufacturing. Moreover, the authors set the scene to prepare the reader for the proposed solution. Namely,

- **II** and **VI** compare secured streaming to conventional secure file storage and transfer and sets out four concrete arguments for streaming: having a thin client on the AM machine and moving calculations to the cloud is more practical than keeping all the processing on the AM machine; files used by AM machinery are large, and size increases over time with the advancement of AM machinery; AM machines do not need a whole file at once, similar to watching a movie—all frames are not needed simultaneously to watch the film;
- **II** uses analogies from the physical world to explain how the physical limitations of cloud computational process are turned into a defense strategy;
- **II** compares secured streaming with with a public/private key encryption approach;
- in **II**, section II-C and Section III-D, we discuss extensively why public-key encryption schemes are insufficient.

The related work of **II** is separated into six aspects of secured communication:

- overall cloud security risks, requirements and mitigation;
- point-to-point and point-to-multipoint secured communication;
- cloud secured file storage and streaming;
- DRM;
- video streaming;
- 3D model streaming.

More detailed overview of related work of the paper **II** can be found in section 4 of this thesis.

The proposed approach of **II** is described in in an abstract way, comparing symmetric and asymmetric-key encryption with the proposed key-less, byte-less encryption. Deeper aspects are explained, e.g., sender and receiver synchronization [45, 59, 61, 124] (in the most basic case, data integrity is based on time function, however there are more options to synchronize multiple sites participating in the secured streaming described in the Publication II, Section IV, Subsection A, Sub-subsection 1), computational bandwidth overhead [16, 23, 26, 27, 36, 38, 39, 41, 42, 48, 50, 52, 73, 94, 99] and cryptographic salt. Then, utilization of physical limitations of the computational process, *live matrix* (initially introduced in **IV**) and proactive and passive cloud nodes paradigms are explained in **II** to support the logic of the abstract solution. The protocol and implementation are also described in **II**.

In **II**, we describe a security evaluation performed with a threat modeling and security analysis based on several threat frameworks [20–22, 95].

A comparison of security solutions is usually performed based on security evaluation and, e.g., listing attack vectors that the solution is prone to. In the publication **II**, we systematically list attack vectors in Section VI of publication II and perform threat modeling and security analysis. It is feasible to compare the proposed solution with any state-of-the-art solution based on the list of attack vectors and threat modeling.

The performance evaluation described in **II** involves three types of test:

- secured streaming between two cloud machines;
- secured streaming between the cloud node and a data consumer node, emulating an AM machine;
- measurement of overhead for different file sizes.

Experiments showed, the proposed method in the paper **II** calculates hashes for a 3-dimensional *live matrix* of 256^3 at an average of 14 revisions per second, and one revision every 5 minutes for a bigger matrix of 4096^3. As a result, the proposed method in **II** is lossless, tightly coupled with secured storage, keeps the data in partitions, has security controls, no single point of failure, and is suitable for peer-to-peer data transfer and bi-directional secured data streams.

Moreover, the proposed solution has been extensively studied. We discuss the number of active users, the number of manufacturing machines connected, and the number of parts produced in Publication **I**, Section 4, namely "the 3DPrinterOS cloud has more than 84.000 users who have generated over three million CAD designs and machine codes. Users have produced more than 950.000 physical parts on 28.000 3D printers in 100 countries".

A more in-depth comparison of the proposed solution with any state-of-the-art solution based on the list of attack vectors and threat modeling is a part of our future work.

5.3 Pattern Recognition of Illegal Designs in 3D Printing

Publication **III** aims to protect IP in AM via an intelligent cyber-physical system for detecting copyright infringements. Publication **III** reviews existing methods to protect AM-related IP (see section 4 for more detail) and defines a novel method for recognizing patterns of illegal objects, including firearms.

A discussion of previous work and a good review of the literature were carried out in the paper **III**. Also, the object recognition process has been well described in the paper **III**.

In related work of the paper **III**, the authors review existing methods to protect AM-related IP. Please see section 4 for more details.

Initially we started with standard algorithms of KNIME [24] machine learning and data mining package. We tried these standard models and algorithms on our data:

- Naive Bayes (Leaner + Predictor);
- SOTA (Leaner + Predictor);
- Fuzzy Rules (Leaner + Predictor);
- MLP Neural Network (Leaner + Predictor);
- PNN Neural Network (Leaner + Predictor);
- Decision Tree (Leaner + Predictor);
- Boosting (Leaner + Predictor);
- Association Rule (Leaner + Predictor);
- SVM (Leaner + Predictor).

Errors were calculated using Scoring and Entropy Scoring. With the standard algorithms we found that the error rate is very high, and achieved on average only 68 percent of correct predictions.

The method proposed in **III** utilizes pattern recognition based on a seriated matrix of 3D designs and their important parameters. The approach has these important steps:

- normalize 3D designs;
- calculate general and intrinsic metadata;
- define typical illegal designs;
- cluster 3D designs into loose groups without strict differentiation;
- challenge new 3D designs against the pattern matrix;
- update the pattern matrix to take into account new designs.

Such a system can not be attacked by adding some zero-function features to plagiarism to cheat the classifier. The system is prone to zero-function feature addition, as we can analyze not only the design as a whole but also based on different features the design has. We explain a similar implementation described in Publication **V**, clause 44. For Publication **V**, we developed a tool that allows us to select different design parts and mark those as IP protected.

The evaluation of the method of the paper **III** was performed on Azure cloud [7] with 5000+ designs in STL format [104] collected from multiple sources [2, 13, 14, 33]. We have implemented the conformity calculation [69, 119] and matrix seriation [67, 68] using Hadoop map/reduce [118]. The authors established experiments on real data with optimal configurations. The results presented in **III** show the method's ability to detect firearm barrels and parts that are not clearly a firearm, though there is a risk of getting false positives if the part is very similar to a part of a firearm, e.g., a bolt with a similar diameter. In future research, we would perform classification on a bigger dataset of 2.14M files and extend the approach presented in this work to a fast 3D design search.

6 Conclusion and Implications for Further Research

In this thesis, we have addressed evolving critical problems of personal manufacturing: democratization, security, IP protection, and safety. We proposed the next step in the evolution of user participation in manufacturing—*personal manufacturing*. We have introduced a novel and a self-sufficient software ecosystem for *personal manufacturing*. Moreover, we have contributed a secure and dependable infrastructure and architecture for AM that leverages the physical limitations of the computational process into a defense strategy that makes distributed file storage and transfer highly secure. We have extended security controls to support IP protection and detect the manufacturing of illegal physical objects using pattern recognition.

In future research, we will continue to work on the cloud manufacturing operating system 3DPrinterOS and deepen investigation in the following areas:

- Implement and perform an experiment for virtual factories, with the aim of allowing any individual to create a virtual factory in minutes to produce any product anywhere using just a web browser.
- Introduce the next generation of AM security by securing and streaming encoded physical impulses from the cloud directly to segmented parts of an AM machine.
- Implement and test protective and detective controls for illegal object detection on the micro-controller level at the edge nodes of the manufacturing ecosystem.

The next steps described above and the corresponding experiments will help to fortify the phenomenon of a smart cyber-physical system for personal manufacturing. It will widen the machine types supported to any type of AM machine, allowing the creation of truly virtual factories. It will also deepen the security layer and comply with the NIST Cybersecurity Framework: Identify, Protect, Detect, Respond, and Recover [107].

Ultimately, this smart cyber-physical system should allow the secure collaboration of multiple different personal manufacturing participants to collaborate on a common goal, e.g., building a car, helicopter, or airplane. Such a system should be able to rearrange its resources to easily produce a different product. This capability would create a fourth industrial revolution.

References

[1] Android (operating system). https://en.wikipedia.org/wiki/Android_(operating_system). Accessed: 09-06-2020.

[2] Defense distributed. https://defcad.com. Accessed: 29-09-2019.

[3] History of operating systems. http://osdata.com/kind/history.htm. Accessed: 23-02-2020.

[4] History of operating systems. https://en.wikipedia.org/wiki/History_of_operating_systems. Accessed: 23-02-2020.

[5] Klipper 3d. https://www.klipper3d.org/Features.html. Accessed: 16-07-2019.

[6] Materialise magics. https://www.materialise.com/en/software/magics. Accessed: 15-07-2019.

[7] Microsoft azure. http://azure.microsoft.com. Accessed: 25-09-2019.

[8] Ms-dos. https://en.wikipedia.org/wiki/MS-DOS. Accessed: 09-06-2020.

[9] Repetier-host. https://reprap.org/wiki/Repetier-Host. Accessed: 16-07-2020.

[10] Slm280 2.0. https://www.slm-solutions.com/en/products/machines/slmr280-20/. Accessed: 15-07-2019.

[11] Stl (file format). https://en.wikipedia.org/wiki/STL_(file_format). Accessed: 23-02-2020.

[12] Stratasys objet1000 plus. https://www.stratasys.com/3d-printers/objet1000-plus. Accessed: 15-07-2019.

[13] Thingiverse. https://www.thingiverse.com. Accessed: 29-09-2019.

[14] Youmagine. https://www.youmagine.com. Accessed: 28-09-2019.

[15] 3D Control Systems, Inc. 3dprinteros cloud world statistics. https://cloud.3dprinteros.com/dashboard/##world-statistics, 2019. [Online; accessed 09-June-2019].

[16] R. Anderson and C. Manifavas. Chameleon-a new kind of stream cipher. In *International Workshop on Fast Software Encryption*, pages 107–113. Springer, 1997.

[17] A. Astovasadourian, O. Naro, and V. Cabanel. 3-d printing protected by digital rights management, Apr. 20 2017. US Patent App. 14/950,431.

[18] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.

[19] H. Badhani, A. Chopra, N. P. Goel, and A. S. Panda. Method and apparatus for controlling printability of a 3-dimensional model, Oct. 4 2016. US Patent 9,457,518.

[20] A. Behl. Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation. In *Information and communication technologies (WICT), 2011 world congress on*, pages 217–222. IEEE, 2011.

[21] A. Behl and K. Behl. An analysis of cloud computing security issues. In *Information and Communication Technologies (WICT), 2012 World Congress on*, pages 109–114. IEEE, 2012.

[22] A. Behl and K. Behl. Security paradigms for cloud computing. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on*, pages 200–205. IEEE, 2012.

[23] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, et al. Sosemanuk, a fast software-oriented stream cipher. In *New stream cipher designs*, pages 98–118. Springer, 2008.

[24] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

[25] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Keccak sponge function family main document. *Submission to NIST (Round 2)*, 3(30), 2009.

[26] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius. Rabbit: A new high-performance stream cipher. In *International Workshop on Fast Software Encryption*, pages 307–329. Springer, 2003.

[27] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 450–466. Springer, 2007.

[28] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow. Elliptic curve cryptography in practice. In *International Conference on Financial Cryptography and Data Security*, pages 157–175. Springer, 2014.

[29] J. Bosch. From software product lines to software ecosystems. In *Proceedings of the 13th international software product line conference*, pages 111–119. Carnegie Mellon University, 2009.

[30] D. P. Bovet and M. Cesati. *Understanding the Linux Kernel: from I/O ports to process management*. " O'Reilly Media, Inc.", 2005.

[31] G. Brunette, R. Mogull, et al. Security guidance for critical areas of focus in cloud computing v2. 1. *Cloud Security Alliance*, pages 1–76, 2009.

[32] M. Bucicoiu, M. Ghideu, and N. Tăpus. Secure cloud video streaming using tokens. In *RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, 2014*, pages 1–6. IEEE, 2014.

[33] E. Buehler, S. Branham, A. Ali, J. J. Chang, M. K. Hofmann, A. Hurst, and S. K. Kane. Sharing is caring: Assistive technology designs on thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 525–534. ACM, 2015.

[34] T. Bui. Analysis of docker security. *arXiv preprint arXiv:1501.02967*, 2015.

[35] M. Butler. Android: Changing the mobile landscape. *IEEE pervasive Computing*, 10(1):4–7, 2010.

[36] P.-L. Cayrel, G. Hoffmann, and M. Schneider. Gpu implementation of the keccak hash function family. In *International Conference on Information Security and Assurance*, pages 33–42. Springer, 2011.

[37] Z. Chen, H. Yin, C. Lin, and L. Ai. 3d-wavelet based secure and scalable media streaming in a centralcontrolled p2p framework. In *Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on*, pages 708–715. IEEE, 2007.

[38] L. Dadda, M. Macchetti, and J. Owen. The design of a high speed asic unit for the hash function sha-256 (384, 512). In *Proceedings of the conference on Design, automation and test in Europe-Volume 3*, page 30070. IEEE Computer Society, 2004.

[39] J. Daemen, L. Knudsen, and V. Rijmen. The block cipher square. In *International Workshop on Fast Software Encryption*, pages 149–165. Springer, 1997.

[40] S. Davies. Microsoft and 3dprinteros announce software bundle to simplify adoption of 3d printing. *TCTMagazine.com*, May 2018.

[41] C. De Cannière. Trivium: A stream cipher construction inspired by block cipher design principles. In *International Conference on Information Security*, pages 171–186. Springer, 2006.

[42] P. Ekdahl and T. Johansson. A new version of the stream cipher snow. In *International Workshop on Selected Areas in Cryptography*, pages 47–61. Springer, 2002.

[43] A. Essop. Hİdromek reduces costs of machinery end-use parts with loop pro 3d printer. *3D Printing Industry*, May 2020.

[44] N. Figay, P. Ghodous, M. Khalfallah, and M. Barhamgi. Interoperability framework for dynamic manufacturing networks. *Computers in Industry*, 63(8):749–755, 2012.

[45] R. Fowler, B. Kotick, and R. Elliott. Polarization analysis of natural and artificially induced geomagnetic micropulsations. *Journal of Geophysical Research*, 72(11):2871–2883, 1967.

[46] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16. ACM, 2016.

[47] D. Glasgow, M. B. MacLaurin, C. E. Sherman, and D. Ramadge. Digital rights and integrity management in three-dimensional (3d) printing, Mar. 14 2017. US Patent 9,595,037.

[48] J. D. Golić. Cryptanalysis of alleged a5 stream cipher. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–255. Springer, 1997.

[49] J. R. Goodman. Using cache memory to reduce processor-memory traffic. *ACM SIGARCH Computer Architecture News*, 11(3):124–131, 1983.

[50] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw. The led block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 326–341. Springer, 2011.

[51] M. Hadzic, E. Chang, and T. Dillon. Methodology framework for the design of digital ecosystems. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 7–12. IEEE, 2007.

[52] M. Hell, T. Johansson, and W. Meier. Grain: a stream cipher for constrained environments. *IJWMC*, 2(1):86–93, 2007.

[53] M. Holland, C. Nigischer, J. Stjepandić, and C. Chen. Copyright protection in additive manufacturing with blockchain approach. *Transdisciplinary Engineering: A Paradigm Shift*, 5:914–921, 2017.

[54] J.-U. Hou, D. Kim, W.-H. Ahn, and H.-K. Lee. Copyright protections of digital content in the age of 3d printer: Emerging issues and survey. *IEEE Access*, 6:44082–44093, 2018.

[55] J. K. Isbjörnssund and A. Vedeshin. Optimized virtual 3d printing build tray allocation, Feb. 19 2015. WO Patent App. WO2015022572A2.

[56] K. Isbjornssund and A. Vedeshin. Method and system for enforcing 3d restricted rights in a rapid manufacturing and prototyping environment, Feb. 27 2014. US Patent App. 13/973,816.

[57] K. Isbjörnssund and A. Vedeshin. Secure streaming method in a numerically controlled manufacturing system, and a secure numerically controlled manufacturing system, Dec. 3 2015. US Patent App. 14/761,588.

[58] M. G. Jaatun, G. Zhao, and S. Alapnes. A cryptographic protocol for communication in a redundant array of independent net-storages. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 172–179. IEEE, 2011.

[59] J. A. Jacobs. *Geomagnetic micropulsations*, volume 1. Springer Science & Business Media, 2012.

[60] J. C. Jensen, D. H. Chang, and E. A. Lee. A model-based design methodology for cyber-physical systems. In *2011 7th International Wireless Communications and Mobile Computing Conference*, pages 1666–1671. IEEE, 2011.

[61] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams. Quantum clock synchronization based on shared prior entanglement. *Physical Review Letters*, 85(9):2010, 2000.

[62] S. Kalpakjian. *Manufacturing engineering and technology*. Pearson Education India, 2001.

[63] Z. C. Kennedy, D. E. Stephenson, J. F. Christ, T. R. Pope, B. W. Arey, C. A. Barrett, and M. G. Warner. Enhanced anti-counterfeiting measures for additive manufacturing: coupling lanthanide nanomaterial chemical signatures with blockchain technology. *Journal of Materials Chemistry C*, 5(37):9570–9578, 2017.

[64] G. Kozmetsky and P. Yue. *The Economic Transformation of the United States, 1950-2000: Focusing on the Technological Revolution, the Service Sector Expansion, and the Cultural, Ideological, and Demographic Changes*. Purdue University Press, 2005.

[65] R. Landauer. Fundamental physical limitations of the computational process. *Annals of the New York Academy of Sciences*, 426(1):161–170, 1984.

[66] Y. Li, D. Li, W. Cui, and R. Zhang. Research based on osi model. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 554–557. IEEE, 2011.

[67] I. Liiv. *Pattern discovery using seriation and matrix reordering: A unified view, extensions and an application to inventory management*. TUT Press Tallinn, 2008.

[68] I. Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 3(2):70–91, 2010.

[69] I. Liiv, A. Vedeshin, and E. Täks. Visualization and structure analysis of legislative acts: a case study on the law of obligations. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 189–190, 2007.

[70] J. H. Lim, B. Panda, and Q.-C. Pham. Improving flexural characteristics of 3d printed geopolymer composites with in-process steel cable reinforcement. *Construction and Building Materials*, 178:32–41, 2018.

[71] N.-H. Lin, T.-H. Huang, and B.-Y. Chen. 3d model streaming based on jpeg 2000. *IEEE Transactions on Consumer Electronics*, 53(1), 2007.

[72] S.-H. Liu, H.-Y. Yu, J.-Y. Wu, J.-J. Chen, J.-L. Liu, and D.-H. Shiue. A secured video streaming system. In *System Science and Engineering (ICSSE), 2010 International Conference on*, pages 625–630. IEEE, 2010.

[73] P. Mahajan and A. Sachdeva. A study of encryption algorithms aes, des and rsa for security. *Global Journal of Computer Science and Technology*, 2013.

[74] V. Manasa and M. Vikram. A secured adaptive mobile video streaming and efficient social video sharing in the clouds. *International Journal of Computer Science and Information Technologies(IJCSIT)*, 5(4):5153–5156, 2014.

[75] K. Manikas and K. M. Hansen. Software ecosystems–a systematic literature review. *Journal of Systems and Software*, 86(5):1294–1306, 2013.

[76] A. Manu, J. K. Patel, S. Akhtar, V. Agrawal, and K. B. S. Murthy. Docker container security via heuristics-based multilateral security-conceptual and pragmatic study. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pages 1–14. IEEE, 2016.

[77] S. Mastorakis. *Peer-to-Peer Data Sharing in Named Data Networking*. PhD thesis, UCLA, 2019.

[78] S. Mastorakis, A. Afanasyev, Y. Yu, and L. Zhang. ntorrent: Peer-to-peer file sharing in named data networking. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–10. IEEE, 2017.

[79] T. D. Mattingly, D. G. Tovey, and J. J. O'brien. System and methods for three dimensional printing with blockchain controls, Sept. 13 2018. US Patent App. 15/913,382.

[80] D. Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239):2, 2014.

[81] E. Miller, D. Long, W. Freeman, and B. Reed. Strong security for distributed file systems. In *Performance, Computing, and Communications, 2001. IEEE International Conference on.*, pages 34–40. IEEE, 2001.

[82] M. Molitch-Hou. Dremel and 3dprinteros partner for 3d printing in the cloud. *Engineering.com*, Jan 2017.

[83] C. Mota. The rise of personal fabrication. In *Proceedings of the 8th ACM conference on Creativity and cognition*, pages 279–288, 2011.

[84] MyStemKits. Mystemkits – 3d printing for stem education. `http://www.mystemkits.com`, 2019. [Online; accessed 17-October-2019].

[85] A. Norta, A. Vedeshin, H. Rand, S. Tobies, A. Rull, M. Poola, and T. Rull. Self-aware agent-supported contract management on blockchains for legal accountability.

[86] C. Okwudire, S. Huggi, S. Supe, C. Huang, and B. Zeng. Low-level control of 3d printers from the cloud: A step toward 3d printer control as a service. *Inventions*, 3(3):56, 2018.

[87] B. O'Neal. Mystemkits & 3dprinteros partner to make 3d printing education streamlined & user friendly. *3DPrint.com*, September 2015.

[88] H. Panetto and A. Molina. Enterprise integration and interoperability in manufacturing systems: Trends and issues. *Computers in industry*, 59(7):641–646, 2008.

[89] K. Rankin. Hack and: what's new in 3d printing, part iv: Octoprint. *Linux Journal*, 2015(257):5, 2015.

[90] S. R. Ray and A. Jones. Manufacturing interoperability. *Journal of Intelligent Manufacturing*, 17(6):681–688, 2006.

[91] Resa Wearables Inc. Resa wearables - 3d printed footcare. `http://www.resakiosk.com`, 2019. [Online; accessed 17-October-2019].

[92] C. Rong and W.-C. Kim. Effective storage security in incompletely trusted environment. In *null*, pages 432–437. IEEE, 2007.

[93] A. Roy. Education and 3d printing - 3dprinteros. `https://www.youtube.com/watch?v=TtOTjFfkcUE`.

[94] R. A. Rueppel. Stream ciphers. In *Analysis and Design of Stream Ciphers*, pages 5–16. Springer, 1986.

[95] P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 280–288. IEEE, 2010.

[96] S. Saunders. Robo to integrate 3dprinteros software so educators can easily adopt 3d printing in the classroom. *3DPrint.com*, May 2018.

[97] S. Saunders. Thanks to new partnership, 3dprinteros software will now power kodak portrait 3d printers. *3DPrint.com*, May 2018.

[98] S. Saunders. 3dprinteros partnering with millebot to containerize large-scale 3d printing. *3DPrint.com*, January 2020.

[99] B. Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *International Workshop on Fast Software Encryption*, pages 191–204. Springer, 1993.

[100] P.-M. Sepp, A. Vedeshin, and P. Dutt. Intellectual property protection of 3d printing using secured streaming. In *The Future of Law and eTechnologies*, pages 81–109. Springer, 2016.

[101] J. Siderska and K. S. Jadaan. Cloud manufacturing: a service-oriented manufacturing paradigm. a review paper. *Engineering Management in Production and Services*, 10(1):22–31, 2018.

[102] S. Smith, G. C. Smith, R. Jiao, and C.-H. Chu. Mass customization in the product life cycle. *Journal of Intelligent Manufacturing*, 24(5):877–885, 2013.

[103] W. R. Stevens and T. Narten. Unix network programming. *ACM SIGCOMM Computer Communication Review*, 20(2):8–9, 1990.

[104] I. Stroud and P. Xirouchakis. Stl and extensions. *Advances in Engineering Software*, 31(2):83–95, 2000.

[105] F. Tao, Y. Cheng, L. Zhang, and A. Y. Nee. Advanced manufacturing systems: socialization characteristics and trends. *Journal of Intelligent Manufacturing*, 28(5):1079–1094, 2017.

[106] F. Tao, L. Zhang, V. Venkatesh, Y. Luo, and Y. Cheng. Cloud manufacturing: a computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(10):1969–1976, 2011.

[107] N. Teodoro, L. Gonçalves, and C. Serrão. Nist cybersecurity framework compliance: A generic model for dynamic assessment and predictive requirements. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 418–425. IEEE, 2015.

[108] A. Tibaut, D. Rebolj, and M. N. Perc. Interoperability requirements for automated manufacturing systems in construction. *Journal of intelligent manufacturing*, 27(1):251–262, 2016.

[109] E. W. Tischhauser. *Mathematical aspects of symmetric-key cryptography*. PhD thesis, Ph. D. thesis, Katholieke Universiteit Leuven, 2012.

[110] P. Trakadas, T. Zahariadis, H. Leligou, S. Voliotis, and K. Papadopoulos. Analyzing energy and time overhead of security mechanisms in wireless sensor networks. In *2008 15th International Conference on Systems, Signals and Image Processing*, pages 137–140. IEEE, 2008.

[111] Ultimaker BV. Ultimaker cura. `https://ultimaker.com/software/ultimaker-cura`, 2019. [Online; accessed 17-October-2019].

[112] Ultimaker BV. Ultimaker cura. https://ultimaker.com/3d-printers/ultimaker-2-plus, 2019. [Online; accessed 17-October-2019].

[113] E. Umaras and M. S. Tsuzuki. Additive manufacturing-considerations on geometric accuracy and factors of influence. *IFAC-PapersOnLine*, 50(1):14940–14945, 2017.

[114] A. Vedeshin. Advanced information retrieval from web pages. In *BCS IRSG Symposium: Future Directions in Information Access 2007*, pages 1–6, 2007.

[115] A. Vedeshin, J. M. U. Dogru, I. Liiv, D. Draheim, and S. Ben Yahia. A digital ecosystem for personal manufacturing: An architecture for cloud-based distributed manufacturing operating systems. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, MEDES '19, page 224–228, New York, NY, USA, 2019. Association for Computing Machinery.

[116] A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. *IEEE Access*, 2019.

[117] A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. Smart cyber-physical system for pattern recognition of illegal 3d designs in 3d printing. In *Communications in Computer and Information Science*, pages 74–85. Springer International Publishing, 2020.

[118] J. P. Verma, B. Patel, and A. Patel. Big data analysis: recommendation system with hadoop framework. In *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, pages 92–97. IEEE, 2015.

[119] L. Võhandu. Fast methods in exploratory data analysis. *Transactions of TTU*, 705:3–13, 1989.

[120] X. V. Wang and X. W. Xu. An interoperable solution for cloud manufacturing. *Robotics and computer-integrated manufacturing*, 29(4):232–247, 2013.

[121] WASP. 3d printed houses for a renewed balance between environment and technology. https://www.3dwasp.com/en/3d-printed-houses-for-a-renewed-balance-between-environment-and-technology/, 2017. Accessed: 25-02-2020.

[122] S. J. Wee and J. G. Apostolopoulos. Secure scalable video streaming for wireless networks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 4, pages 2049–2052. IEEE, 2001.

[123] C. Xenakis, N. Laoutaris, L. Merakos, and I. Stavrakakis. A generic characterization of the overheads imposed by ipsec and associated cryptographic algorithms. *Computer Networks*, 50(17):3225–3241, 2006.

[124] M. Xu, D. A. Tieri, E. Fine, J. K. Thompson, and M. J. Holland. Synchronization of two ensembles of atoms. *Physical review letters*, 113(15):154101, 2014.

[125] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[126] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18, 2010.

[127] D. Zhou. Research on 3d printed creations through course design for the democratisation of production: Interdisciplinary opportunities for steam education. In *5th International STEM in Education Conference: Post-Conference Proceedings*, pages 523–530, 2019.

Acknowledgements

First, I would like to thank my supervisors, Assoc. Prof. Innar Liiv and Prof. Dirk Draheim for help, support, stimulative discussions, opinions, feedback, and this challenging but enriching research experience, and much more.

Second, I would like to thank John Dogru for being a friend and a business partner with whom we built together 3DPrinterOS.

Third, I would like to thank all investors who invested in 3D Control Systems, Inc.

Fourth, I would like to thank all my family members, especially my wife Jevgenia and two sons Ivan and Maxim, for help, support and patience while doing research, writing this thesis, and while I and John all these years were building the world's first operating system for 3D printers.

Fifth, I would like to thank my parents for supporting and cultivating my will to study and invent things and gifting me my first personal computer.

Finally, I would like to thank our country, Estonia, for providing free education and an amazing startup and business environment.

Abstract
Smart Cyber-Physical System for Personal Manufacturing

Today, society reached an inflection point in automated manufacturing - the era of *personal manufacturing*. Now, home users, small, medium, and Fortune 2000 enterprises use 3D printers, CNC mills, water, and laserjets, and other types of robotics to manufacture products at the point and time of need. The inter-connectivity and ease of use became very important. In addition to that, today, a manufacturing file contains all the needed intellectual property to produce a production-grade physical part. Individuals and companies need to secure this intellectual property from end-to-end. Another important implication is to protect society from manufacturing an obviously harmful object like firearms, weapons, ammunition.

However, 3D printers today are at the same stage of development as computers were 60 years ago. Every 3D printer hardware manufacturer is trying to develop its software, which creates a steep learning curve for the end-user and a fragmentation of the market. The world underwent multiple operating system consolidations, first the consolidation of personal computers' operating systems with UNIX's adoption, then the majority of smartphone manufacturers adopted Android. We have analyzed the most influential related work looking at the topic from different angles. There is a clear gap in the industry; there is no end-to-end, secured standard across different manufacturers, easy to use operating system for 3D printers.

The contribution of this thesis is a Smart Cyber-Physical System for Personal Manufacturing that addresses the described problems. We believe the future of manufacturing is one secure, standardized platform for automated manufacturing, which cures multiple vendors' software fragmentation, removes usability friction for the end-users, and protects the intellectual property rights and protects the society from harmful parts.

In this thesis, we have contributed three building blocks of a Smart Cyber-Physical System for Personal Manufacturing. 1. A novel and a self-sufficient software ecosystem for *personal manufacturing*. 2. A secure and dependable infrastructure and architecture for AM that leverages physical limitations of the computational process into a defense strategy that makes distributed file storage and transfer highly secure. 3. We have extended security controls to support IP protection and detect the manufacturing of illegal physical objects using pattern recognition.

For each of the building blocks of a Smart Cyber-Physical System for Personal Manufacturing, we have carried out the experiments.

1. For the digital ecosystem for 3D printing, we performed two types of experiments: long- and short-term. During a long experiment, users have produced more than a million parts on 32 thousand 3D printers. The short term experiment showed that, on average, it takes five times less effort to produce a physical part using the proposed software ecosystem. 2. For the secure data infrastructure for 3D printing, we performed a security evaluation by performing a threat modeling and security analysis based on several threat frameworks. Performance evaluation was carried out with three types of tests: item secured streaming between two cloud machines; item secured streaming between the cloud node and a data consumer node, emulating an AM machine; overhead for different file sizes. 3. For the pattern recognition of illegal designs in 3D printing, we performed the method evaluation on Azure cloud with 5000+ designs in STL format collected from multiple sources. We have implemented the conformity calculation and matrix seriation using the Hadoop map/reduce [118]. The results showed the ability to detect firearm bar-

rels and parts that are not clearly a firearm. There is a risk of getting false positives if the part is very similar to a part of a firearm, for example, a bolt with a similar diameter.

In this thesis, we have addressed evolving critical problems of the advanced manufacturing industry: software democratization, security, IP protection, and safety. We proposed the next step in the evolution of user participation in manufacturing—*personal manufacturing*, which can not sufficiently exist without the discussed smart cyber-physical system for personal manufacturing.

Kokkuvõte
Tark küberfüüsikaline süsteem personaalseks tootmiseks

Täna jõudis ühiskond automatiseeritud tootmise pöördepunkti - *personaalse tootmise* ajastusse. Nüüd kasutavad kodukasutajad, väikesed ja keskmise suurusega ettevõtted ning Fortune 2000 ettevõtted 3D-printereid, CNC-veskeid, vee- ja laserjugasid ning muud tüüpi robootikat toodete valmistamiseks. Ühenduvus ja kasutusmugavus on muutunud väga oluliseks. Lisaks sisaldab tootmisfail kogu vajalikku intellektuaalomandit tootmiskõlbliku füüsilise osa tootmiseks. Eraisikud ja ettevõtted peavad selle intellektuaalse omandi otsast lõpuni ise endale tagama. Teine oluline mõte on kaitsta ühiskonda selliste ilmselgelt kahjulikkude esemete nagu relvade ja laskemoonade valmistamise eest.

3D-printerid on tänapäeval samas arengujärgus kui arvutid 60 aastat tagasi. Iga 3D-printeri tootja proovib oma tarkvara välja töötada, mis loob kasutajate hulgas segadust ja lõhestab turgu. Maailmas tehti mitu operatsioonisüsteemide konsolideerimist, esmalt personaalarvutite operatsioonisüsteemide konsolideerimine UNIX-i kasutuselevõtuga, see-järel kasutasid enamus nutitelefonide tootjaid Androidi. Oleme analüüsinud sellega seo-tud kõige mõjukamat tööd, vaadeldes teemat erinevate nurkade alt. Tööstuses on selge lõhe; erinevate tootjate jaoks puudub otsast lõpuni turvaline standard, 3D-printerite jaoks hõlpsasti kasutatav operatsioonisüsteem.

Selle lõputöö panuseks on tark küberfüüsikaline süsteem personaalseks tootmiseks, mis tegeleb kirjeldatud probleemidega. Usume, et tootmise tulevik on üks turvaline, standardiseeritud platvorm automatiseeritud tootmiseks, mis parandab mitme müüja tarkvara killustatust, eemaldab lõppkasutajatele kasutatavuse vastuolu ning kaitseb intellektuaalomandi õigusi ja kaitseb ühiskonda kahjulike osade eest.

Selles lõputöös oleme panustanud personaalse tootmise targa küberfüüsikalise süsteemi kolme ehitusplokki. 1. Uudne ja isemajandav tarkvaraökosüsteem *personaalseks tootmiseks*. 2. Autmatiseeritud tootmise turvaline ja töökindel infrastruktuur ja arhitektuur, mis kasutab arvutusprotsessi füüsilisi piiranguid kaitsestrateegiana, mis muudab hajutatud failide salvestamise ja edastamise üliturvaliseks. 3. Meil on laiendatud turvakontroll, et toetada intellektuaalomandi kaitset ja tuvastada ebaseaduslike füüsiliste objektide tootmine mustrituvastuse abil.

Igas eelnevas teemas oleme läbi viinud katsed Tark Küberfüüsikalise süsteemi personaalseks tootmiseks. 1. 3D-printimise digitaalse ökosüsteemi jaoks viisime läbi kahte tüüpi katseid: pikaajalised ja lühiajalised. Pika katse jooksul on kasutajad tootnud 32 tuhande 3D-printeri abil üle miljoni detaili. Lühiajaline eksperiment näitas, et kavandatud tarkvara ökosüsteemi abil füüsilise osa tootmiseks kulub keskmiselt viis korda vähem jõupingutusi. 2. 3D-printimiseks mõeldud turvalise andmete infrastruktuuri jaoks viisime läbi turvalisuse hindamise, viies läbi ohu modelleerimise ja turvalisuse analüüsi, mis põhines mitmel ohuraamistikul. Toimivuse hindamine viidi läbi kolme tüüpi testidega: üksuse turvaline voogesitus kahe pilvemasina vahel; üksuse turvaline voogesitus pilvesõlme ja andmetarbija sõlme vahel, jäljendades AM-masinat; erinevate failisuuruste üldkulud. 3. Ebaseaduslike kujunduste mustrite tuvastamiseks 3D-printimisel viisime läbi meetodi hindamise Azure'i pilves koos 5000+ STL-vormingus kujundusega, mis olid kogutud mitmest allikast. Oleme rakendanud vastavuse arvutamise ja maatriksi seotuse, kasutades Hadoopi Map/Reduce. Tulemused näitasid võimet tuvastada relvatünne ja osi, mis ei ole selgelt tulirelv. On oht saada valepositiivseid tulemusi, kui osa on väga sarnane tulirelva osaga, näiteks sarnase läbimõõduga polt. Selles lõputöös oleme käsitlenud töötleva tööstuse arenevaid kriitilisi probleeme: tarkvara demokratiseerimine, turvalisus, intellektuaalomandi kaitse ja ohutus.

Pakkusime välja järgmise sammu kasutajate tootmises osalemise arengus — *personaalne tootmine*, mida ei saa piisavalt eksisteerida ilma arutatud targa küberfüüsikalise süsteemita, mis on vajalik personaalseks tootmiseks.

Appendix 1

I

A. Vedeshin, J. M. U. Dogru, I. Liiv, D. Draheim, and S. Ben Yahia. A digital ecosystem for personal manufacturing: An architecture for cloud-based distributed manufacturing operating systems. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, MEDES '19, page 224–228, New York, NY, USA, 2019. Association for Computing Machinery

A Digital Ecosystem for Personal Manufacturing: An Architecture for Cloud-based Distributed Manufacturing Operating Systems

Anton Vedeshin, John Mehmet Ulgar Dogru
3D Control Systems, Inc.
San Francisco, California, USA
anton;john@3DPrinterOS.com

Innar Liiv, Dirk Draheim, Sadok Ben Yahia
Tallinn University of Technology
Tallinn, Estonia
innar.liiv;dirk.draheim;sadok.ben@taltech.ee

ABSTRACT

Recently, we have witnessed the advent of personal manufacturing, where home users, small, medium, and Fortune 500 enterprises use devices such as 3D printers, CNC mills, and robotics to manufacture products locally. We have been developing a digital ecosystem of personal manufacturing for the last seven years. This ecosystem is currently used or being tried by 111 Fortune 2000 enterprises. In this paper, we focus on the creation of the cloud-based manufacturing operating system, 3DPrinterOS, to address an evolving critical problem of personal manufacturing. We introduce a novel software ecosystem architecture to sustain a massive communication load of command, control, and telemetry data to and from millions of manufacturing machines and users. Our solution allows users to create and deploy their own applications into 3DPrinterOS cloud operating system. Our long term experiments show that over the last five years, 95, 000 users have generated over three million CAD designs and machine codes, and produced more than 1, 030, 000 physical parts on 32, 000 manufacturing machines in 100 countries. Short term experiments showed that, on average, it is five times faster to perform a 3D print using 3DPrinterOS.

CCS CONCEPTS

• **Applied computing** → **Enterprise computing infrastructures**; • **Computer systems organization** → **Cloud computing**; • **General and reference** → *Experimentation*; • **Information systems** → Enterprise applications.

KEYWORDS

3DPrinterOS SECO, personal manufacturing, cloud manufacturing, cloud operating system, cloud manufacturing operating system, digital ecosystem architecture

ACM Reference Format:
Anton Vedeshin, John Mehmet Ulgar Dogru and Innar Liiv, Dirk Draheim, Sadok Ben Yahia. 2019. A Digital Ecosystem for Personal Manufacturing: An Architecture for Cloud-based Distributed Manufacturing Operating Systems. In *11th International Conference on Management of Digital EcoSystems (MEDES '19), November 12–14, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3297662.3365792

MEDES '19, November 12–14, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-6238-2/19/11.
https://doi.org/10.1145/3297662.3365792

1 INTRODUCTION

The impressively fast adoption of automated manufacturing (AM) technologies such as 3D printers, CNC mills, and robotics indicates that this novel approach to manufacturing can become a key enabler for the real-time economy of the future, i.e., represent a possible paradigm shift in manufacturing towards personal manufacturing. In such a paradigm, people will not buy a ready-made product at the factory, but obtain raw material and produce products locally, utilizing their own or publicly available AM machinery.

We have been developing 3DPrinterOS, a digital ecosystem for personal manufacturing, for the last seven years. It is currently deployed or in trials at 111 of the enterprises from the Forbes 2000 list. During this journey, we have faced challenges with interoperability, usability, scalability, and network connection stability, among others while building a self-sufficient AM ecosystem.

In the following, we briefly summarize the main contributions of this paper:

(1) Discuss the motivation behind our creation of a cloud-based manufacturing operating system, 3DPrinterOS, to address an evolving critical problem of personal manufacturing;
(2) Propose the next step in the evolution of user participation in manufacturing—*personal manufacturing*;
(3) Propose a unique and a self-sufficient software ecosystem for *personal manufacturing*. Such a system would support all of the components necessary to produce a physical object from a digital representation of an idea, under either automatic or user control, allowing users to move from an idea to a physical object in one click;
(4) Introduce a novel, cloud-based software ecosystem capable of sustaining a massive communication load of command, control, and telemetry data coming to and from millions of manufacturing machines and users. Our solution allows users to create and deploy their own applications in a cloud operating system.

The remainder of the paper is organized as follows: In Section 2, we discuss the related work and our motivation to create the 3DPrinterOS software ecosystem (SECO). In Section 3, we describe the system overview, which includes the architecture and functions. In Section 4, we discuss a five-year experiment and one short-term experiment. In Section 5, we conclude the paper and suggest directions for future work.

2 RELATED WORK AND MOTIVATION

We have analyzed the most influential and highly cited works published on cloud manufacturing [17], social aspects of advanced manufacturing, and interoperability. This section provides context for our work in the field. We start by focusing on the social aspects of manufacturing.

2.1 Social aspects of manufacturing

In their work "Advanced manufacturing systems: socialization characteristics and trends," F. Tao et al. [20] analyze the degree and scope of resource sharing since the 1960s. Then, they describe four phases in the evolution of manufacturing resource sharing, which they say happens: a) within an enterprise; b) among enterprises; c) among industries and across regions; d) in society as a whole. Moreover, they [20] describe the degree of user participation in the manufacturing of a product: 1) Buy—the role of the user is minimal; the user buys a ready-made mass produced product. There is no interaction between the consumer and the manufacturer. A good example is the Ford Model T. 2) Buy and choose—the user has a chance to choose a more satisfying product from among a greater variety of products that are "mass customized" [18]. The manufacturer performs market research to segment their customers, and provides each segment with a product customized to match their preferences. For example, a car manufacturer such as Toyota produces different models for each customer segment. Within each segment, customers can select interior and exterior colors, engine power, and optional equipment. 3) Buy, choose, and design—in addition to the above, the consumer participates in the design of the product. The manufacturer produces a customized product for each user; for example, personalized 3D-printed insoles for shoes manufactured based on 3D scans of the consumer's feet. 4) Full customization—in addition to the above, the user can monitor the manufacturing process online and select and arrange the delivery method and date.

In our work, we seek to contribute the next logical step of the evolution of consumer participation in manufacturing by creating a new model: 5) *Personal manufacturing*—beyond full customization, the consumer is involved not just in monitoring production, but in the actual manufacturing process. The consumer either owns the equipment for automated manufacturing or has easy access to such. The user can select the quality, price, speed, material, production technology, and location of manufacturing; this may include choosing a popular solution, or designing a custom one.

2.2 Cloud manufacturing

In their work, Tao et al. [21], define and compare cloud manufacturing to cloud computing, and name the key advantages of cloud manufacturing as: a) reducing the idle time of manufacturing machinery and increasing utilization; b) greatly reducing the cost of entry for home users, small, medium, and even Fortune 500 enterprises, as it provides immediate access to high-value manufacturing resources (e.g. expensive automated manufacturing machinery) without up-front capital investments.; c) similarly, reducing the cost of ownership via savings on manufacturing infrastructure maintenance and administrative costs and reduced energy use; d) making it easier to scale production and business in line with client demand; e) generating new types of business models and ways to deliver products [26], for example, MyStemKits [11], RESA [14]; f) allowing enterprises and people to focus only on their core business and service rather than the entire manufacturing life cycle.

2.3 Interoperability

In their respective works, Tao et al. [20], Ray et al. [13], Tibaut et al. [22], Wang et al. [25], Panetto et al. [12], and Figay et al. [6] name the interoperability of manufacturing systems and components as one of the most compelling challenges in the evolution of cloud manufacturing resource sharing.

Interoperability requirements affect the architecture of manufacturing cloud operating systems. In this work, we seek to quickly adjust and keep up with new manufacturing machines as they become available. There are currently thousands of different types and modifications of manufacturing machines, and this number continues to increase.

3 SYSTEM OVERVIEW

3.1 Architecture

3DPrinterOS connects users to manufacturing machines (Figure 1). Users have web browsers installed on the devices they use to access the cloud OS. Manufacturing machines, which are industrial IoT devices in this case, are connected to the cloud through firmware or a *cloud client*. Ideally, the 3DPrinterOS firmware is deployed within a manufacturing machine and controls the low-level operations involved in producing parts. If a manufacturing machine does not have enough computing power to connect to the cloud over the network and provide the implementation of the 3DPrinterOS protocol with command and control and telemetry data, then it is connected to external hardware (Linux, Windows or Mac) connected to the cloud with the *cloud client* installed. Both the 3DPrinterOS firmware and cloud client can receive printer profiles, material profiles and slicing profiles, lists of manufacturing files, and projects, and cache these data locally.

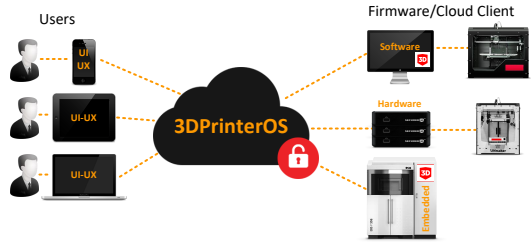


Figure 1: High-level conceptual diagramm of cloud-based distributed manufacturing operating system 3DPrinterOS

The architecture of a cloud operating system consists of three layers: application, libraries, and cloud kernel (Figure 2).

The *application layer* is where 3DPrinterOS provides basic functionality for end users, like file uploads and storage, toolpath visualization, an end-user dashboard, management for print jobs, real-time updates for the user interface (UI), storage manager, authentication of the user, user manager, notification manager, printer

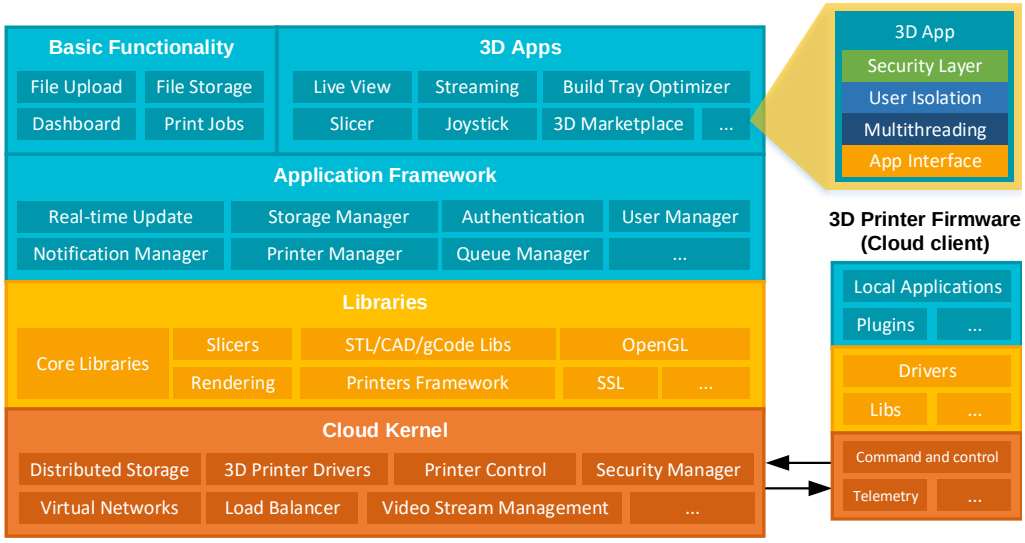


Figure 2: Architecture diagram of cloud-based distributed manufacturing operating system

manager, queue manager, default slicers. An important part of the 3DPrinterOS cloud platform is its *app engine and marketplace*, which allows the deployment of applications developed by third parties—*3D Apps*. Such apps allow users of the platform to perform a very specific niche action. Each *3D App* has a common interface (Figure 2) with a security layer, user isolation, multithreading, and standardized UI. A *3D App* is an encapsulated application, wrapped in a Linux container (e.g. a Docker container [9]). Through encapsulation, we achieve a high level of security [3, 8]: *3D Apps* cannot access the memory space and data of other apps. User isolation guarantees that a separate instance of the application will process each user's data, and after processing, will return the result and be destroyed. Multithreading allows faster performance of operations. The app interface allows *3D App* developers to create a unique experience for the end user.

The *libraries layer* of 3DPrinterOS cloud platform provides numerous core libraries as virtual resources for the application layer including *3D Apps*. Core libraries are: 3D rendering engine, libraries for STL, CAD and gCodes, OpenGL, cryptography frameworks (e.g. key-less, byte-less encryption), and other 3D printer frameworks.

The *cloud kernel layer* is responsible for the most low-level operations in the cloud, such as distributed storage, 3D printer drivers, printer command and control, security manager, virtual networks, load balancer, content distribution network (CDN), and video stream management.

The *3D printer firmware*, which we call the cloud client, also has three layers. Although our cloud OS is not intended to provide extensive functionality on the printer side (as the cloud does), it has minimized versions of some *3D Apps*. It also can receive printer profiles, material profiles and slicing profiles, lists of manufacturing files, and projects, and cache these data locally.

Taken together, the user interface, cloud, and industrial IoT components form a cloud-based distributed manufacturing operating system.

3.2 Roles, Artifacts, Relations

To describe the 3DPrinterOS ecosystem in our work, we have used the Software Ecosystem (SECO) approach formulated by Manicas et al. [7], where "a software ecosystem is the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services. Each actor is motivated by a set of interests or business models and connected to the rest of the actors and the ecosystem as a whole with symbiotic relationships, while, the technological platform is structured in a way that allows the involvement and contribution of the different actors". The proposed ecosystem in this paper belongs to the web operating system-centric ecosystem class according to the software ecosystem taxonomy proposed by Bosch [2].

The roles, artifacts and relations of the 3DPrinterOS SECO are shown in Figure 3. Actors, actor types, actors' contribution to the 3DPrinterOS SECO, and the benefits they receive are described as follows:

Orchestrator is 3DPrinterOS. The orchectrator is neutral to all other actors and responsible for the well-functioning of the ecosystem. The orchestrator develops and manages the cloud platform and other parts of the system, mediates relationships and the value flow among other actors of the ecosystem by settings the rules, processes, business procedures, setting and monitoring quality standards. The orchestrator sustains a base service layer by developing and providing simple high-level applications for end users. In this case, the orchestrator could be compared to the Android [4] operating

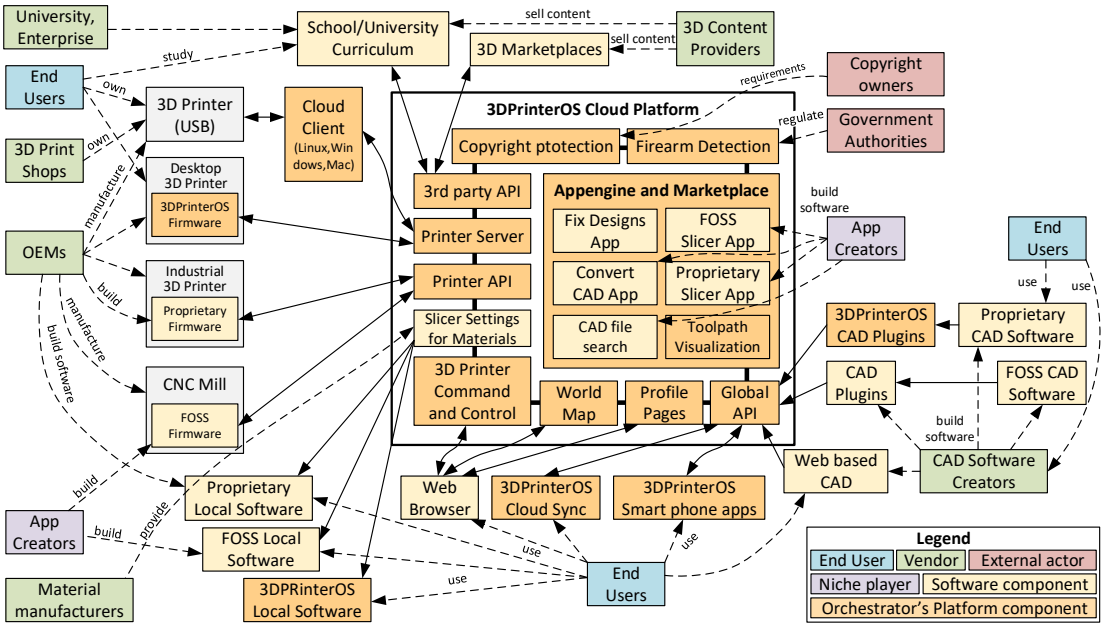


Figure 3: Overview of 3DPrinterOS SECO

system, which provides some simple default apps for end users. However, if the end user needs a more specific application, they can obtain it from the app marketplace.

Niche players are *3D Apps* creators; they contribute to the ecosystem by creating very specific niche applications. For instance, they override the default basic applications on the platform, e.g., 3DPrinterOS has developed the MagicFix application, which checks CAD files for inconsistencies and address them. There are multiple niche players who have developed more specific applications to detect and fix issues in CAD files. These applications provide additional value to the end users by publishing their *3D Apps* on the 3DPrinterOS platform. In other settings, niche players provide the main value, e.g., slicer software for 3D printers. The orchestrator does not have a public version of the slicer, and all the slicer *3D Apps* on the public platform are developed by niche players.

External actors perform activities that are limited to the actor's interest and provide indirect value to the ecosystem by observing the evolution of the ecosystem. For example, government authorities want to make sure that no illegal parts are 3D printed, e.g. firearm parts. Copyright owners want to ensure that parts produced with the means of automated manufacturing (AM) are according with the copyright contracts, and does not infringe creators' rights.

Vendors distribute the products of the ecosystem to end-users or other vendors. The products are bundles of AM hardware and software, vendors' own software bundled with the ecosystem components, complete integration or separate components. In case of the ecosystem presented in this paper vendors are original hardware manufacturers (OEM), AM machine manufacturers and 3D printer manufacturers particularly. Vendors manufacture hardware, integrate their machines with 3DPrinterOS, and benefit from increased sales and number of end users. The other representatives of vendors of the ecosystem are 3D print shops, who own AM machines and connect those to the platform to increase utilization and make more money. Material manufacturers provide precise slicer settings to the platform to improve manufacturing quality, and increase material sales though increased popularity and credibility.

End users or customers are the persons, companies, and other entities that either purchase or obtain a complete or partial SECO [7]. In our case, there are four types of end users:

a) the do-it-yourself community, who benefit from ease of use, and single interface to any AM machine;
b) small and medium businesses, who save time and money managing their fleet of 3D printers, reduce prototyping turnover time, and reduce time to market;
c) educational institutions, who provide fully self-service access to AM machines, reducing the costs to minimum, utilizing the power of data analytics to improve the service and more efficiently procure materials from material manufacturers;
d) Fortune 2000 enterprises, who save at least one human resource per every 10 AM machines, stay IT compliant, and have a full overview of who is manufacturing what in their enterprise on AM machines.

All types of users benefit from utilizing the 3DPrinterOS SECO, regardless of whether they manufacture using their own machinery, or machinery selected from among their organization's machinery, or that of their organization's partners or subcontractors.

Users can create virtual factories from 3DPrinterOS SECO artifacts, e.g. a full process flow, starting from searching for a CAD design and ending with delivery of a manufactured and assembled product without actually owning AM equipment or CAD software.

4 EXPERIMENTS

3DPrinterOS SECO is an experiment by itself. The project was started in 2014 as an experiment to see whether the idea of AM and 3D printing SECO would work. The last five years have showed the success of the SECO; as of today, the 3DPrinterOS cloud platform [1] has more than 95, 000 end users who have generated over three million CAD designs and machine codes. 3DPrinterOS end users have produced more than 1, 030, 000 physical parts on 32, 000 3D printers and other AM machines in 100 countries. These statistics double every six months. 3DPrinterOS SECO components are licensed to vendors including Microsoft [5], Bosch [10], Kodak [16], Robo3D [15] and other popular desktop 3D printer manufacturers, and distributed to their end users.

Moreover, the 3DPrinterOS SECO is implemented by top US universities: Duke, MIT, Purdue, Harvard, Yale, Caltech, and Texas A&M. Students use 3DPrinterOS as a self-service way to manufacture parts for their projects, with access to hundreds of manufacturing machines. All universities involved have reported a large reduction in costs (instead of 1 AM lab technician per 5 to 10 manufacturing machines, on average, to just one person in the lab), an average of $10x$ higher utilization of manufacturing machines, $100x$ more student involvement, and a reduction in waste.

An experiment that was carried out at TalTech [19] as a part of 3D-printing classes for university students.For the first experiment, we selected 74 people, aged 21 to 55 years, 49 men and 25 women. 22 of the experimental group had previously used a 3D printer (Group A). The other 52 had not used a 3D printer before (Group B). Half (Group 1) of each group (A and B) were asked to 3D-print a part using the 3D printing software, Cura [23], native to the 3D printer used— the Ultimaker 2 [24]. The other half (Group 2) were asked to perform the same task using 3DPrinterOS digital ecosystem. For people in groups A1,A2,B1 and B2, it took an average of 10, 2, 42, and 8 min, respectively, to 3D print a design. The results showed, that on average, it was five times faster for members of both groups to print a 3D part using 3DPrinterOS.

5 CONCLUSION

In this paper, we have described 3DPrinterOS SECO—a digital ecosystem for personal manufacturing, which allows users to move from an idea to a physical object in one click. We have proposed and explained the architecture for a self-sufficient cloud-based distributed manufacturing operating system which would allow the user to perform all necessary steps to produce a product at the point and time of need, with zero latency. We have described the most important functions of the system and presented the results of an ease-of-use experiment.

REFERENCES

[1] 3D Control Systems, Inc. 2019. 3DPrinterOS cloud world statistics. https://cloud.3dprinteros.com/dashboard/#world-statistics. [Online; accessed 09-June-2019].

[2] Jan Bosch. 2009. From software product lines to software ecosystems. In *Proceedings of the 13th international software product line conference*. Carnegie Mellon University, 111–119.

[3] Thanh Bui. 2015. Analysis of docker security. *arXiv preprint arXiv:1501.02967* (2015).

[4] Margaret Butler. 2010. Android: Changing the mobile landscape. *IEEE Pervasive Computing* 10, 1 (2010), 4–7.

[5] Sam Davies. 2018. Microsoft and 3DPrinterOS announce software bundle to simplify adoption of 3D printing. *TCTMagazine.com* (May 2018). https://www.tctmagazine.com/3d-software-news/microsoft-3dprinteros-software-bundle-simplify-3d-printing/

[6] Nicolas Figay, Parisa Ghodous, Malik Khalfallah, and Mahmoud Barhamgi. 2012. Interoperability framework for dynamic manufacturing networks. *Computers in Industry* 63, 8 (2012), 749–755.

[7] Konstantinos Manikas and Klaus Marius Hansen. 2013. Software ecosystems–A systematic literature review. *Journal of Systems and Software* 86, 5 (2013), 1294–1306.

[8] AR Manu, Jitendra Kumar Patel, Shakil Akhtar, VK Agrawal, and KN Bala Subramanya Murthy. 2016. Docker container security via heuristics-based multilateral security-conceptual and pragmatic study. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 1–14.

[9] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014, 239 (2014), 2.

[10] Michael Molitch-Hou. 2017. Dremel and 3DPrinterOS Partner for 3D Printing in the Cloud. *Engineering.com* (Jan 2017). https://www.engineering.com/3DPrinting/3DPrintingArticles/ArticleID/14021/Dremel-and-3DPrinterOS-Partner-for-3D-Printing-in-the-Cloud.aspx

[11] MyStemKits. 2019. MyStemKits – 3D Printing for STEM Education. http://www.mystemkits.com. [Online; accessed 17-October-2019].

[12] Hervé Panetto and Arturo Molina. 2008. Enterprise integration and interoperability in manufacturing systems: Trends and issues. *Computers in industry* 59, 7 (2008), 641–646.

[13] Steven R Ray and AT Jones. 2006. Manufacturing interoperability. *Journal of Intelligent Manufacturing* 17, 6 (2006), 681–688.

[14] Resa Wearables Inc. 2019. RESA Wearables - 3D printed footcare. http://www.resakiosk.com. [Online; accessed 17-October-2019].

[15] Sarah Saunders. 2018. Robo to Integrate 3DPrinterOS Software So Educators Can Easily Adopt 3D Printing in the Classroom. *3DPrint.com* (May 2018). https://3dprint.com/234094/robo-integrating-3dprinteros-software/

[16] Sarah Saunders. 2018. Thanks to New Partnership, 3DPrinterOS Software Will Now Power KODAK Portrait 3D Printers. *3DPrint.com* (May 2018). https://3dprint.com/214798/3dprinteros-kodak-portrait/

[17] Julia Siderska and Khair S Jadaan. 2018. Cloud manufacturing: a service-oriented manufacturing paradigm. A review paper. *Engineering Management in Production and Services* 10, 1 (2018), 22–31.

[18] Shana Smith, Gregory C Smith, Roger Jiao, and Chih-Hsing Chu. 2013. Mass customization in the product life cycle. *Journal of Intelligent Manufacturing* 24, 5 (2013), 877–885.

[19] Tallinn University of Technology. 2019. Tallinn University of Technology. https://taltech.ee. [Online; accessed 17-October-2019].

[20] Fei Tao, Ying Cheng, Lin Zhang, and Andrew YC Nee. 2017. Advanced manufacturing systems: socialization characteristics and trends. *Journal of Intelligent Manufacturing* 28, 5 (2017), 1079–1094.

[21] Fei Tao, Lin Zhang, VC Venkatesh, Y Luo, and Ying Cheng. 2011. Cloud manufacturing: a computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 225, 10 (2011), 1969–1976.

[22] Andrej Tibaut, Danijel Rebolj, and Matjaž Nekrep Perc. 2016. Interoperability requirements for automated manufacturing systems in construction. *Journal of intelligent manufacturing* 27, 1 (2016), 251–262.

[23] Ultimaker BV. 2019. Ultimaker Cura. https://ultimaker.com/software/ultimaker-cura. [Online; accessed 17-October-2019].

[24] Ultimaker BV. 2019. Ultimaker Cura. https://ultimaker.com/3d-printers/ultimaker-2-plus. [Online; accessed 17-October-2019].

[25] Xi Vincent Wang and Xun W Xu. 2013. An interoperable solution for Cloud manufacturing. *Robotics and computer-integrated manufacturing* 29, 4 (2013), 232–247.

[26] Ding Zhou. 2019. Research on 3D Printed Creations through Course Design for the Democratisation of Production: Interdisciplinary Opportunities for STEAM Education. In *5th International STEM in Education Conference: Post-Conference Proceedings*. 523–530.

Appendix 2

II

A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. *IEEE Access*, 2019

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Secure Data Infrastructure for Personal Manufacturing Based on a Novel Key-less, Byte-less Encryption Method

ANTON VEDESHIN[1], JOHN MEHMET ULGAR DOGRU[1], INNAR LIIV[2,5], SADOK BEN YAHIA[2,4], and DIRK DRAHEIM[3]

[1]3D Control Systems, Inc., San Francisco, 94129 CA, USA (e-mail: anton; john@3dprinteros.com)
[2]Department of Software Science, Tallinn University of Technology, Akadeemia Tee 15a, 12618 Tallinn, Estonia (e-mail: innar.liiv; sadok.ben@taltech.ee)
[3]Information Systems Group, Tallinn University of Technology, Akadeemia Tee 15a, 12618 Tallinn, Estonia (e-mail: dirk.draheim@taltech.ee)
[4]Faculty of Sciences of Tunis, University of Tunis El Manar, LIPAH-LR11ES14, 2092 Tunis, Tunisia
[5]Centre for Technology and Global Affairs, University of Oxford, Manor Road, Oxford OX1 3UQ, United Kingdom

Corresponding author: Anton Vedeshin (e-mail: anton@3dprinteros.com).

ABSTRACT We are witnessing the advent of personal manufacturing, where home users and small and medium enterprises manufacture products locally, at the point and time of need. The impressively fast adoption of these technologies indicates this approach to manufacturing can become a key enabler of the real-time economy of the future. In this paper, we contribute a secure and dependable infrastructure and architecture for that new paradigm. Our solution leverages physical limitations of the computational process into a defense strategy that makes distributed file storage and transfer highly secure. The main idea is to replace asymmetric or public-key encryption functions with an unkeyed, collision, second preimage, and preimage resistant cryptographic hash function. Such a cryptosystem does not have an inverse function H^{-1}. We challenge each block hash against the full hash table to recreate the original message. To illustrate the approach, we describe secured protocols that provide a number of desirable properties during both data storage and streaming. Similar to proof-of-work blockchain consensus algorithms, we parameterized the solution based on the amount of infrastructure available. Experiments show the proposed method can recalculate hashes for a 3-dimensional *live matrix* of 256^3 at an average of 14 revisions per second, and one revision every 5 minutes for a bigger matrix of 4096^3. The increase in cloud infrastructure cost is insignificant compared to the level of protection offered.

INDEX TERMS Communication system security, computer aided manufacturing, content distribution networks, data security, data storage systems, distributed computing, information security, intelligent manufacturing systems, technology social factors, virtual manufacturing.

I. INTRODUCTION

WE are witnessing the advent of personal manufacturing, where home users, small and medium enterprises use devices such as 3D printers, CNC mills, laser jets, and robotics to manufacture products locally, at the point and time of need. The impressively fast adoption of these technologies strongly indicates that this novel approach to manufacturing can become a key enabler for the real-time economy of the future, i.e., a possible paradigm shift in manufacturing toward personal manufacturing. In such a paradigm, people and organizations would not buy a ready-made product. Instead, they would obtain raw material and produce products using their own or locally accessible automated manufacturing (AM) machinery.

With the growing popularity of AM, robotic process automation (RPA), self-driving cars, automated medical devices, video and hologram streaming and internet of things (IoT) in general, the need to securely store and transfer streamable file types such as machine instructions and manufacturing files becomes more and more important.

Thus, the requirements for a modern secure distributed file storage and transfer are changing, and efficient methods of secured cloud storage and streaming are becoming a compelling need. However, securing cloud file storage and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

transfer is a challenging task [1]. The nature and properties of modern files types impose certain constraints on how secure distributed file storage and transfer methods should operate.

One such constraint is the need to repeatedly access streamable files line by line or layer by layer without inconsistencies, delay, or compromising security through exposure of the whole file at once. In this paper, we address this problem and introduce a possible solution based on an efficient approach that utilizes technical limitations of the cloud and leverages them into a security control and defense strategy.

The main idea is to replace an asymmetric or public-key encryption functions with an unkeyed, collision, second preimage, and preimage resistant cryptographic hash function. Such a cryptosystem does not have an inverse function H^{-1}, and no key to decrypt the hash and get message back unless we pre-calculate a full hash table. We challenge each block hash against the full hash table to recreate an original message. To illustrate this approach, we have constructed secured protocols that provide a number of desirable properties to secure machine codes at rest and during delivery to stream consumption device.

The previous generation [2]–[4] of our solution has been implemented and proven over several years as a mechanism to securely deliver content to 3D printers from the cloud. Today, the 3DPrinterOS cloud has more than 84 000 users who have generated over three million CAD designs and machine codes. Users have produced more than 950 000 physical parts on 28 000 3D printers in 100 countries [2]; these values double every six months [2]. The technology is licensed to Bosch [3], Kodak [4], and other popular desktop 3D printer manufacturers. The solution described in this paper completely reworks the first [5] and second generation [2]–[4] of this secure content delivery mechanism and extends it to any type of manufacturing machine or complex IoT device with command, control, and telemetry.

The main contributions of this paper are: a) a novel, key-less, byte-less encryption method, ready for application to AM; b) an approach that leverages the physical limitations of the computational process [6] into a defense strategy; c) a threat model and security analysis of the proposed approach.

The main use case is the transfer of machine codes from secured cloud storage to a network-connected manufacturing machine. Other potential applications include streaming of a) video; b) holographic video; c) voice communication; d) medical data; e) business file data; f) telemetry, including command and control data to and from self-driving cars.

The remainder of this paper is organized as follows. In Section II, we introduce additional background and discuss the topics addressed in this paper. In Section III, we analyze and discuss why existing cloud file storage and transfer solutions such as digital rights management (DRM), video streaming and 3D model streaming fail to address critical constraints and security problems adequately. In Section IV, we explore a relatively new paradigm of cloud security, *live matrix*, proactive and passive cloud nodes, and our protocol. In Section V, we thoroughly describe the proposed cloud application infrastructure and architecture; in Section VI, we discuss strong and vulnerable points of such an approach. In Section VII, we describe the setup used to evaluate the proposed method by conducting experiments with a local cloud of machines. Finally, Section VIII concludes the paper by summarizing the results and indicating issues to be addressed in future work.

II. SETTING THE SCENE

This section prepares the reader for the proposed solution, which is described starting in Section IV.

A. STREAMING VERSUS CONVENTIONAL SECURE FILE STORAGE/TRANSFER

1) Argument: Importance of machine instructions

Seventy years ago, in the so-called "paper age," most products' technical drawings were prepared on paper. Imagine an attacker obtained pictures of the paper sketches of an innovative product. In the best-case scenario, it took many years to find or even build production technology, train engineers, set up a factory and production lines to produce prototypes and then a real product. In the worst case, there is no way to build the product using copies of the sketches, as the "secret sauce" required to build that product is somewhere down the production line, inside the heads and hands of the engineers working at a specific factory. A good example is rocket fuel; even with all the sketches of rocket structure and shape, people still need to identify and prepare fuel.

About thirty years ago, we entered the digital age, with the use of computer-aided design (CAD), computer-aided engineering (CAE), computer-aided manufacturing (CAM), computer-aided process planning (CAPP), computer-aided quality assurance (CAQ), production planning and control (PPC), and enterprise resource planning (ERP) tools [7]. However, these tools were initially used primarily to create a virtualization of a product to make measurements, manage bill of materials (BOM), and provide simulations to facilitate quicker changes to a product's structure and shape during prototype testing cycles. Much manual work was still required, including post-processing and manual surface finishing. People are accustomed to using very basic solutions, like digital rights management (DRM), to secure CAD/CAM/CAE designs.

In the past, if such a DRM-protected CAD/CAM/CAE design was compromised, the barriers discussed above would still slow the rate of the product's production and distribution. Compared to the "paper age" example, with decades required to produce the product, in the digital age, it might take only six months to figure out the details, find production facilities, and produce a marketable product.

In the personal manufacturing age, CAD/CAM/CAE intended for AM already has the "secret sauce" baked in. In other words, the proprietary information required to produce the market-ready product is inside the file. If such a design is compromised, the attacker can reach the market with a production-quality product in just a few days, if not hours.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

Designs intended for AM and 3D printing contain all of the information needed to manufacture a real production quality product according to exact specifications: make and model of the manufacturing device, direction of layers infill, tolerances, surface finish, materials, speeds, temperatures, durability and taking into account force distribution and dispensation. With recent advances in AM technology, it is possible to manufacture a real working part or a usable product from a CAD/CAM/CAE design in just a few hours.

2) Argument: An AM machine is a thin client

The amount of information contained within modern CAD/CAM/CAE files for AM creates a load on the whole supporting infrastructure and requires substantial computing power. There is no way to put a supercomputer into each AM machine.

Over time, there has been a trend in AM to move as much calculation to the cloud as possible due to the low cost of cloud computing power. Initially, slicing for 3D printers was performed on the workstation built into a 3D printer (e.g., [8], [9]). Then, slicing software moved to engineers' workstations [10]. Now, slicing has moved to the cloud [2], with machine code streamed to the AM machine.

The next important step is to stream stepper motor pulses from the cloud directly to the AM machine. Firmware is moving to the cloud. As with software and faster computing, this move improves hardware operation, with incredible increases in quality and speed. For instance, Okwudire et al. [11] sent a low-level stepper motor commands from a server to simplified firmware, which interpreted simple commands and proxied them to the stepper motor drivers. They measured an increase in printing quality and speed.

AM machines should have a thin client built in, not a workstation [8], [9]. This thin client will interpret commands and send back current status and metrics. If the AM does not achieve a certain temperature or speed, the cloud needs to know, to update its manufacturing execution system (MES) and users about the delay. This approach will reduce costs and eliminate the need for local software updates. Moreover, the increase in calculation complexity possible in the cloud enables faster, smoother operation of local AM machines.

To explain why, we must first outline the basic steps that every contemporary AM machine firmware performs: a) read machine code into memory; b) interpret machine code into movements between coordinates; c) plan path through coordinates; d) calculate accelerations and decelerations with lookaheads taking into account inertia and potential forces; e) project movements to the stepper motor axis; f) ensure the motors and toolhead follow the programmed trajectory.

It is difficult to achieve excellent manufacturing quality when performing such processing on microcontrollers. Most firmwares perform only minimal prediction of the toolhead path. As a result, movement of the toolhead creates excessive vibration and noise, and it sometimes hits the wall of the machine. These phenomena cause drops in manufacturing quality with any increment in manufacturing speed, despite the machines' excellent and frequently over-engineered hardware. The problem hides in the microcontrollers, which spend most of their computing time calculating trajectory. The less computing time the microcontroller spends on planning, and the more on operating the hardware, the better the manufacturing speed and quality.

To move the toolhead one millimeter, a stepper motor must perform a certain number of steps. For example, a 0.9 degree per step stepper motor performs a whole revolution in 25 full steps [12]. Such a motor will produce torn movements and generate substantial vibration. Moreover, the movements will be slow because of the inability to accelerate and decelerate efficiently; if configured to operate at high speeds, the machine will skip steps, resulting in missing manufacturing tolerances and overall lower product quality. The same motor operated with so-called micro-stepping, set at 1/32 of a step, will move much more smoothly, but require 800 steps per revolution [12]. However, not every microcontroller can maintain this rate of feeding steps into the motor driver. For context, an ATmega 16 MHz microcontroller with Marlin firmware achieves fewer than 10 000 steps per second (10 kHz) [13].

Moving path planning out of the firmware to a nearby computer increases manufacturing speed and quality. This was achieved by a team of researchers behind the Klipper project [14]. The same ATmega 16 MHz microcontroller described above, but operated with Klipper firmware [14], achieves 151 000 steps per second (151 kHz). It also drives the motors more smoothly, with fewer errors, and improved manufacturing quality. In the Step Benchmarks table [14] we can see that the same hardware can be 10x more efficient with the right software and more computing power. To achieve such improvements, we will ultimately stream encoded physical signal commands from the cloud to AM machines. The method proposed in this paper is ready for these types of applications.

3) Argument: Large file sizes

To explain why AM machine codes should be streamed versus downloaded and stored, we will use the example of a very simple 3D design—an annular cylinder—created in OpenSCAD software [15], [16].

The file for a given object will have a different size depending on which stage of manufacturing it is prepared for, and will involve different representations of the 3D object. We have depicted data file sizes at different stages of digital design for automated manufacturing in Fig. 1. As it shows, file size increases exponentially when moving from a less systematically specified representation of the object to the more specific representation needed to produce the production part.

In Step 1, the initial CAD design can be a few lines of code to mathematically represent a part. In Step 2, the STL file prepared for manufacturing is a set of triangles in space representing a CAD file; in addition to the overall shape of the object it contains information on manufacturing tolerances,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

the higher is precision the bigger is the file size. The lower the tolerances, the bigger the file size. In Step 3, machine code is produced from that file; this code is specific to a certain AM machine make and model. In addition to the shape of the object, it contains information about each individual layer the 3D printer will build to create the object. Each layer requires a certain number of movements of the toolhead. Each movement has an associated speed and information about the amount and speed of material extrusion. In Step 4, the command sequence for stepper motors file represents all of the signals that go to the stepper motor driver to execute the machine code. It includes calculations of acceleration and deceleration, takes into account inertia, timing, and many other factors. This is the exact recipe for how the part is produced. Changes in this last stage of preparation will affect the tolerances, quality, and speed of manufacture.

4) Argument: The whole file is not needed at once

In a past experiment [17], we found that a CAD file of a computed tomography (CT) scan of the human brain required about 2 GB, the corresponding medium-quality AM machine codes 6 GB, and print time for the full-size brain was 96 h. For a high-quality 3D print of the human brain, the machine code would be 36 GB, requiring approximately two weeks of manufacturing time on a 3D printer. The 3D printer did not need the entire file at once, as the manufacturing process takes time, and it was possible to transfer the file in smaller segments.

B. PHYSICAL LIMITATION OF COMPUTATIONAL PROCESSES

This is a basic example explaining how the physical limitation of computational process and different types of bottlenecks can be turned into a defense strategy in the cloud.

Let's use an analogy from the physical world. Let distilled water represent data we want to protect. A bottle of distilled water is put on a table, see Fig. 2 a). One approach to obtain the water without opening a lock on the bottle is to drill a small hole to let the water leak out (Fig. 2 b). Our storage solution could be compared with constantly changing bottles, and a robot which pours water from one bottle to another, adding and removing chemicals using different chains of chemical reactions to protect the water (Fig. 2 c). In this scenario, the water that is actually poured is, for example, sometimes a different acid, sometimes a different alkali. An attacker can still start drilling a hole in the bottle, but the bottle is still and steady only for a minute before the robotic arm starts to pour it to a different glass and add some other chemicals to change the state of the liquid. Only robot knows how many chemical transformations and in what sequence would lead back to the original distilled water.

If an attacker starts to drill holes into the bottles to steal the liquid, that attack requires time. If drilling a hole takes 5 minutes, and the bottle is only available in steady condition for 1 minute, then this is a clear bottleneck—a physical limitation. Now imagine a hacker used a faster way to drill a hole. It still takes time and there is a physical limitation—the diameter of the hole (in our approach, the network connection between the nodes and between the hacked node and secured cloud node). Now, the attacker starts to get a liquid. But if it takes, say, an hour to obtain all liquid from the bottle it will do the attacker no good—*this exact bottle holds the liquid for only 1 minute*—before the bottle is changed and the physical composition of the liquid is changed. The attacker has obtained some small amount of an unknown liquid, with no information about how to turn it back into its original form. By drilling subsequent holes and getting smaller amount of liquids at different stages of the chemical chain or recipe, the attacker will end up with a mysterious mixture with a complex chemical composition. The attacker will not know how to turn this mixture back into its original form. The attacker may have substantial time and computational power to analyze the liquid and to use brute force to get the original mixture. But this is near-impossible, as at some later time even the robot will not know what happened in the past; it does not have enough storage to keep versions of all of the obsolete recipes and chemical reactions. The faster the robot performs its manipulations, the harder it is to access the bottle for a reasonable amount of time, to drill holes or pump out the contents of the bottle.

Now, how does the solution described above translate to a computer problem? The metaphor described above with robotic arms and chemicals in bottles explains that it is hard to steal information that is constantly moving and transforming. This is the physical limitation. We compare our metaphoric example with what our solution does in Fig. 2:

1) A bottle with water and a lock on the lid **to** data store with data at rest, encrypted with a key (Fig. 2 a).
2) A drill bit, a key-ring with different master keys and lock picks, and a hacksaw attacking the locked bottle with water **to** encrypted data at rest and the use of various attack vectors to get to the data at rest (Fig. 2 b). This comparison represents an encrypted file in storage. Once an attacker gets a copy of the storage or the file, cracking it is only a matter of time.
3) Robotic arm **to** *live matrix* (Fig. 2 c). Our solution shuffles the data faster than an attacker can download it from the cloud, due to the physical limitations of computer systems, for example, the network interface.
4) Bottle with added chemicals **to** the data state in our solution (Fig. 2 c). The data state is static for a short amount of time, then it is changed. Within this short time period, it is hard to successfully extract the full file. The attacker ends up with partial data extracted at different states.
5) Broken bottles **to** expired data states (Fig. 2 c). If the data state is expired and not yet removed from the computer memory, it can no longer be used for retrieval of the data; thus, attacking it does not help crack the data store.
6) Drill bit **to** attack vector (Fig. 2 c). Any attack vector

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

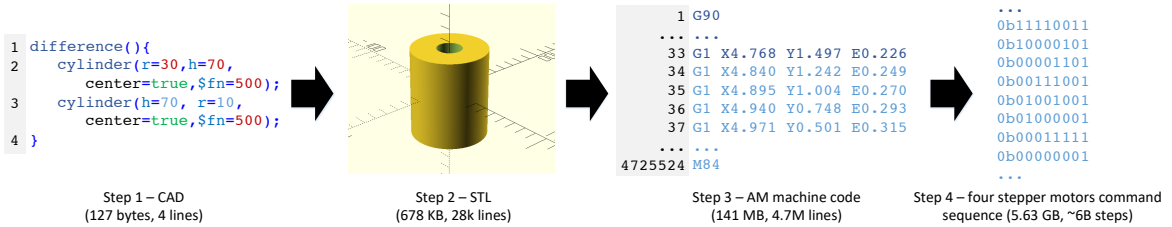


FIGURE 1. File size at different stages of digital design for automated manufacturing: From left: CAD design; an STL file prepared for manufacturing; machine codes for a specific AM machine make and model; command sequence for AM machine stepper motors. File size exponentially increases from the less systematically specified representation of the object to the more specific representation required to produce the physical part.

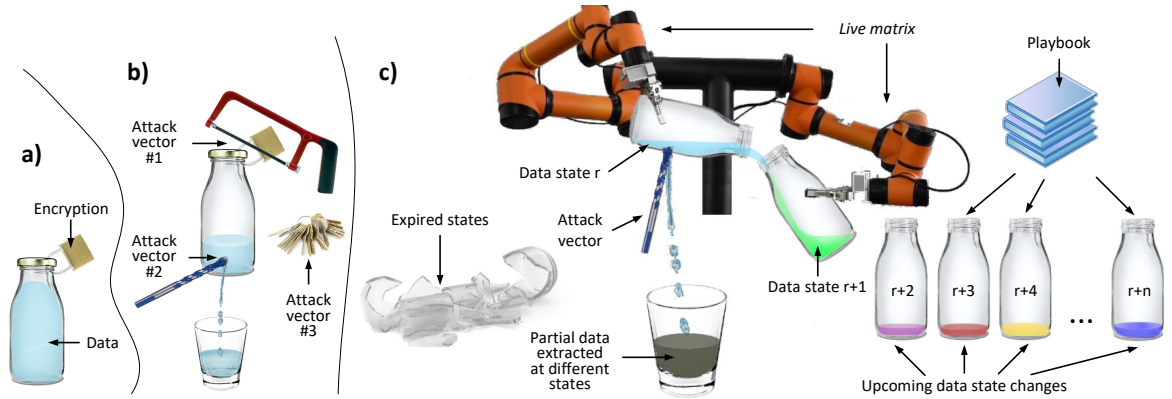


FIGURE 2. Representations of a) data encrypted with a static key; b) three attack vectors on static key encryption; c) dynamic key encryption with constantly changing data states—the state changes more quickly than the time required to physically extract the data.

requires some amount of time to extract data. Before an attacker can extract the data, its state becomes obsolete, and the attack must be started from the beginning. Any attack through a computer system will face a physical limitation if the secured storage uses these physical limitations as a defense mechanism.

7) The bottle with the next chemical solution where the robotic arm pours the current chemical solution **to** current data state r and next data state $r + 1$ (Fig. 2 c).
8) Queue of bottles with different chemical solutions according to a recipe **to** upcoming data states $r + 2, r + 3, r + 4, ..., r + n$ according to the *playbook* (Fig. 2 c).

C. PUBLIC/PRIVATE KEY ENCRYPTION

Why not simply use public/private key encryption to protect manufacturing files? This approach is unfortunately prone to attacks in a manner similar to DRM. If a manufacturing file is encrypted with a static key, and the file is transferred and collected by the attacker, then decrypting it is only a matter of time.

One approach could use software like network security research tool Fiddler [18]. Fiddler can receive encrypted traffic using public-key, e.g., HTTPS traffic. When installed on a machine, it collects all dynamic public/private keys for all communication to/from that computer. It is relatively trivial to use an approach like this to collect dynamic keys and decrypt the files being transferred, without even compromising the software receiving the file. To compromise our solution, Fiddler would need to understand the in-memory *live matrix* data structure, understand how it is being calculated, and only then potentially perform an attack. This is a much more complicated scenario to execute compared to public-key encrypted file transferred over HTTPS.

III. RELATED WORK

In this section, we present work that we consider to be close to the requirements described above and categorize relevant papers into six subcategories for a more systematic discussion. We start with some general considerations of cloud security, and then go more deeply into specific solutions, like point-to-point and point-to-multipoint secured communication, cloud secured storage, DRM, video streaming, and 3D streaming.

A. CLOUD SECURITY RISKS, REQUIREMENTS, AND MITIGATION

In [19], Brunette et al. provide a comprehensive analysis of possible issues in cloud security and how to mitigate them. They present a solid approach to assess existing cloud applications and provide a requirement base for the design of secure cloud solutions. That work provides notable recom-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

mendations. However, from our perspective, a next level—an integral solution—is necessary. For the sake of an ultimate security solution for cloud storage and file transfer, we need a change in philosophy, and a new paradigm—*live matrix*—which we describe in Section IV-C.

B. POINT-TO-POINT AND POINT-TO-MULTIPOINT SECURED COMMUNICATION

We examined related research on peer-to-peer, point-to-point and point-to-multipoint communication. First, most such solutions tend to use lower layers of the OSI model, mostly layer 3, the network layer. This positively affects the speed and throughput of the communication. At the same time it makes most of the protocols proprietary and exotic, which may make them hard to widely implement for AM machines. In contrast, the solution we propose in this paper is network layer and protocol agnostic, as the only information that is transferred is cryptographic hashes. Our solution would benefit from using a lower layer of OSI model, and streaming hashes over a lower level of the OSI model is a topic worthy of future research and experiments.

Second, the main efforts in the literature are focused on resolution of peers and finding and re-routing if a peer is disconnected. These mechanisms can compliment the solution described in this paper. Many approaches to point-to-point and point-to-multipoint communication security employ basic private/public key encryption, which does not prevent the exposure of intellectual property.

Mastorakis et al. [20], [21] discuss peer-to-peer file sharing application designs and implementations that run on top of Named Data Networking (NDN). The security aspect is in the nature of the NDN architecture; however, this suggests cryptographically signing every packet in the network. NDN uses a distribution of data encryption keys as encrypted NDN data. Because it implements security at the protocol level, NDN offers good protection against negligence, in contrast to TCP/IP, where applications are responsible for security. Although NDN is considered to be the future of Internet [22], it is still at the stage of work in progress, and not yet ready for full production grade implementation.

C. CLOUD SECURED FILE STORAGE AND STREAMING

In their cryptographic protocol [23], Jaatun et al. present an approach that is similar to ours. They segment files among the redundant array of independent net-storages in the computing cloud. The main thrust of their solution is the distribution of data across different cloud providers. Thus, the individual data deposits do not expose enough information about the owner and the file to make them vulnerable. In addition, in order to return the file to the user, the data must be reassembled. In our approach, we similarly distribute file parts to many machines in the cloud; however, we do not set a specific constraint on the form and number of cloud providers; our approach can utilize physical computing machines, virtual machines, Docker containers from one or several providers, etc.

Miller et al. [24] propose several robust security schemes for distributed file systems. They use segmentation of files into file blocks, and file block encryption with asymmetric keys. Similar to [24], we split a file into segments and encrypt each segment with its own key. But we go beyond this, and propose a continuous re-encryption of file segments, with constantly changing keys. Moreover, we may constantly re-encrypt the symmetric keys that data segments are encrypted with. In our approach, re-encryption happens constantly on all cloud nodes at a preset file, computational, or cost limit.

In [25], Giuseppe et al. describe improved proxy re-encryption schemes for keys and apply them to secured distributed storage. We apply a similar approach in our solution, but to file segments, and not just keys. Furthermore, we re-encrypt continuously, regardless of reads and writes to storage. Cloud computing infrastructure prices drop each year; thus, such a re-encryption approach is feasible for use with millions of files.

D. DIGITAL RIGHTS MANAGEMENT

There are many practical DRM-like approaches that are widely used in cloud storage and transfer. These include ECFS [26], and others mentioned in the same paper. In DRM, a file is usually encrypted using a symmetric or asymmetric key or a key combination before it is stored or transferred. In order to access the file, the data consumer needs the key. When an attacker obtains the key by, for example, buying the protected content once, brute force, social engineering, etc., then the file can be used or redistributed infinitely. DRM methods are usually lightweight and can be functional without any need for intensive cloud computing power. From our perspective, DRM methods are too vulnerable by their nature (Sec. II-C).

E. VIDEO STREAMING

Numerous existing streaming approaches [27]–[31] work efficiently and consistently for video and music. Even though some of the protocols have consistency checks, they are not expected to deliver every single byte; insignificant data loss or delay caused by network problems is expected. However, this could be an issue for sensitive data, like CAD designs. For example, in the case of streaming designs to automatic manufacturing machines such as 3D printers or CNC mills, data transfer should be consistent and lossless: loss of a single byte while streaming is unacceptable, as this can lead to a AM machine malfunction or a defective product. At the same time, the streaming should be highly secured, which is not usually a requirement for media streaming protocols. In this paper, we show how to securely stream encrypted file segments directly from a highly secure distributed file storage.

F. 3D MODEL STREAMING

In [32], Lin et al. describe a method to encode 3D models into a JPEG stream in order to transfer 3D designs. However, the solution is not comprehensive and has clear limitations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

In prior research [33], we theoretically described *live matrix* as a paradigm applied to secured 3D content delivery. Our prior work is purely theoretical, and so lacks technical details and a real implementation of the method. This paper's contribution is to extend the initial idea with the necessary details for implementation and to technically broaden it to any type of secure file storage and transfer. Furthermore, we describe a threat model and conduct a thorough security analysis. It is worth mentioning that we eliminate the transcoding of files for streaming introduced in a prior work [5], [34].

In previous work [5], [34], we have explained in detail the necessity for secured streaming of 3D files and discussed methods to enforce 3D file copyrights. Our previous approach targeted a small niche case to secure 3D design transfer to 3D printers. That solution is very machine code-centric and lacks a tight coupling with the secured storage. Furthermore, it is vulnerable at the point of extracting a 3D design from the storage and re-encoding it for streaming. In the current paper, we propose a much more secure and consistent end-to-end method to store and stream files—regardless of file type—and without the need to re-encode the file for streaming.

IV. PROPOSED APPROACH

Relying on the principles and paradigms described below, we describe a working solution for highly secure distributed file storage and transfer.

A. ABSTRACT SOLUTION

For the cryptosystem [35]

$$D_{\mathrm{d}}(E_{\mathrm{e}}(m)) = m \tag{1}$$

where E is an encryption function, e is an encryption key, D is a decryption function, d is a decryption key, and m is a message, if $d = e$, then we have symmetric encryption. However, if d does not equal e, we have a public-key or asymmetric-key cryptosystem. The main feature of this cryptosystem is that only knowledge of the *static* decryption key is required to decrypt the message.

For unkeyed cryptographic hash function H, which is collision, second preimage, and preimage resistant [35]

$$H(m) = h \tag{2}$$

there is no such inverse function H^{-1}, and no key d, e to decrypt the hash h and get message m back:

$$H^{-1}(h) \in \varnothing. \tag{3}$$

In other words, a key for a hash does not exist.

Then, the only way to retrieve the original message is to hash all possible combinations and compare the hashes one by one. For example, if we know that the original message is five symbols from the ASCII table [36], given a strong cryptographic hash function [37] the only way to obtain the original message is to look the hash up in the table—the so-called brute force method [38]. To achieve this, we would need to create a hash table with ${}_{256}P_5$, a trillion elements, and then look up the original message by the hash. This makes a brute force attack impractical, requiring substantial computational power.

Our solution is based on the complexity of retrieving the original message by its hash. To make the methods work, the task of our solution is to keep the complexity of the potential message set within a certain threshold, so just enough computing power is available to perform the calculations required.

The solution relies on a logic similar to that behind RSA SecurID tokens [39]. In that case, the same function with the same cryptographic seed is running on both the RSA server and the token (a small piece of hardware with a battery) in a user's pocket. In order to log in to the system, the user must enter a username, password and the code from the RSA token. The code on the RSA token expires every minute, and a new code is generated and is shown on the screen. A minute later, when the code expires, there is no way to reuse the code. In the proposed solution, we do something similar, but by recalculating a hash table and parts of the file on a regular basis—for example, every minute. After a minute, another hash table is calculated to accommodate file parts; the previous hash table expires and is deleted from memory. The process iterates over and over again.

In an abstract way, the solution works like this:

1) A is the (finite) set of symbols from ASCII table;
2) S is the (finite) set of file segments;
3) Each file segment s is set to a fixed length of m bytes;
4) t is a time variable and k is cryptographic salt.
5) G is the (finite) set of permutations of A set members with sample size m, so that ${}_{\mathrm{A}}P_{\mathrm{m}} \in G$.
6) *Sender and receiver side:* for each member g of a set G, together with time t and salt k, we calculate a corresponding hash $h_{\mathrm{g_n,t,k}}$ using hash function H. The hash is stored in a hash table T_{t} along with the original member g.

$$H(g,t,k) = h_{\mathrm{g,t,k}} \rightarrow g \in T_{\mathrm{t}}. \tag{4}$$

7) *Sender and receiver side:* when time t is incremented, table T_{t} expires at the moment $t' = t + \Delta t$; Step 6 is repeated, and a new table $T_{\mathrm{t+1}}$ is calculated, so

$$T_{\mathrm{t'}} \in \varnothing; T_{\mathrm{t'}} \neq T_{\mathrm{t+1}} \tag{5}$$

8) *Sender side:* for each member s of a set S we look up a corresponding hash $\mathrm{h_{g,t,k}}$ in table T using function L and send the hash to the receiving device. At this point we call hash $\mathrm{h_{s,t,k}}$ a $hint$. This $hint$ does not contain any actual bytes from set S, and there is no key as such to decrypt the $hint$ (see Eq. 3):

$$L(s, T_{\mathrm{t}}) = h_{\mathrm{s,t,k}}. \tag{6}$$

9) *Receiver side:* When the $hint$ arrives, the receiver challenges it. The receiver performs a lookup against the local version of table T using function L. If such an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

element is found, L returns a file segment s; otherwise, the value is undefined:

$$L(hint, T_t) = f; L(hint, T_{t'}) \in \varnothing. \quad (7)$$

10) The successfully received file is a set of $hints$ positively challenged against the hash tables $T_t, T_{t+1}, ..., T_{t+n}$.

In step 6, on the AM machine side, the same hash table with the same potential elements of set G should be generated in advance, taking into account exactly the same timing and salt (like RSA SecurID tokens have the same time-based function running on the server and the hardware token).

In step 9, when a $hint$ arrives on the AM machine side, we look it up in the current hash table, and retrieve (or do not retrieve) the corresponding file segment. We recreate a file from successfully found segments. This is not a decryption function in terms of 1, as there is no key as such in terms of that equation, and actual static bytes are not transferred in its terms:

$$T_t \not\equiv d; Hint \not\equiv c \quad (8)$$

$$\Rightarrow L(hint, T_t) = f \not\equiv D_d(c) = m \quad (9)$$

$$\Rightarrow L(L(s, T_t), T_t) = f \not\equiv D_d(E_e(m)) = m \quad (10)$$

In the case of TLS/SSL, the actual encrypted bytes of the file are transferred. In our solution, only $hints$, which expire, are transferred. It is not possible to get a real byte of the file based on that $hint$ a minute later. This is similar to the way in which an expired RSA SecurID code cannot be used.

In the next three sub-subsections we will explain important considerations about our approach.

1) Sender and receiver synchronization

Our approach is agnostic to synchronization method. Time t could be logical or physical time; in our experiments, we use physical time (UTC). Distributed machines can synchronize time against time servers. A minor change in time would not usually put the sender and receiver out of sync, unless the difference is larger than the *live matrix* state expiration time. For high latency networks or situations when time is slightly out of sync the *live matrix* expiration time could be increased so there is always a previous *live matrix* state available. At the same moment, there are two *live matrices* available—the current one and a previous one.

Dependency on time could be removed completely if we synchronize against other sources. Future accessible synchronization methods, might include natural phenomena like geomagnetic micropulsations [40], [41], seismic activity, gravity, blockchain block number, shared prior quantum entanglement [42], [43], and others.

2) Computational and bandwidth overhead

Overall, cipher text has a positive difference in length between encrypted text and plain text.

Our solution has a bigger positive overhead compared to well-known stream ciphers [44]–[50], block ciphers [51]–[54] and plain text in terms of computation and bandwidth.

Commonly, in stream ciphers [55] the cipher text length has an insignificant positive overhead compared to plain text. A key is used to generate a stream, which is then combined with the plain text to get the cipher text using an XOR operation. This does not significantly affect the amount of information transferred nor computation needed, as XOR is computationally inexpensive.

In block ciphers [56], padding is frequently added to plain text to make it equal to the block size, increasing the bandwidth overhead. Block ciphers also have computational overhead to encrypt each block compared to plain text.

Our solution has a parametric trade-off between computational complexity and security. By increasing the *live matrix* recalculation frequency, we increase the security level as well as the calculation complexity.

Our solution bandwidth overhead depends on the selected *live matrix* size and cryptographic hash function. The closer the number of bytes m to the output number of bytes of the hash function, the lower the bandwidth overhead. The recommended hash function output length should be close to the m, but not smaller than m. Overhead can be calculated as:

$$overhead = length(h_{s,t,k}) - length(m) \quad (11)$$

so that

$$length(h_{s,t,k}) \geqslant length(m). \quad (12)$$

The computational overhead of our approach is lower than that of block ciphers, and depending on the stream cipher algorithm, can be even smaller than stream ciphers. Hash function calculation is less expensive than AES and DES [57], [58]. Another possibility, which is highly application-dependent (e.g., not very practical for IoT and self-driving cars), is to scale hash calculation using a GPU and ASIC implementation of hash functions [59], [60]. However, block and stream encryption is difficult to implement using a GPU and ASIC-based approach.

3) Cryptographic salt

Salt k is not static. It changes with time, and could be an access code for a one-time manufacturing license, a PIN code, or part of a private key. Further, parameters other than k affect the setup; even if k is compromised, an attacker would still need to figure out the algorithmic setup and parameters.

B. PHYSICAL LIMITATIONS OF THE COMPUTATIONAL PROCESS

Total security does not exist. Breaking into any system is just a matter of the time and money required to exploit its weaknesses. Indeed, cloud computing itself processes huge amounts of data in parallel, a capability that can be used against attacks. However, the storage, network and computing power of the cloud have physical limits to writing and readings files, transferring files over the network, calculating hashes, and encrypting or decrypting information. The philosophy behind our solution is to set an attacker versus

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

a computing cloud and leverage the physical limitations of the computational process [6] as security controls. Similar to proof-of-work blockchain consensus algorithms [61], we parameterize the solution based on the amount of available infrastructure. The more computational power used, the harder and more expensive it becomes to carry out an attack.

Henceforth, we consider a hacker as a human individual, a group of hackers with special tools, or an automated script or bot with sufficient computing power. A hacker can never know all parameters and exact details of our secured cloud implementation, and it will take a considerable amount of time to find and exploit these weaknesses. This could be mitigated with detective cybersecurity strategies. If the hacker is equipped with comparable computing power, then the physical limitations of the computational process come into play.

There is physical latency at all levels of hardware and software during computational processes. In order to reduce latency, computer L1 and L2 cache memory is located very close to the processor [62]. The more distant some resource is from the processor, the higher the latency. For example, a network interface is usually a main bottleneck for distributed systems [63]. The operating system limit of open ports and I/O descriptors in Linux can be a bottleneck [64]. Our approach is to use these limitations and bottlenecks and turn them into a defense strategy.

C. LIVE MATRIX

Live matrix is a multidimensional data structure in which the data is constantly changing state. The state may even change millions of times per time frame, Δt_n, depending on the computing resources allocated. We refer to a different state during a certain time frame $\Delta t_1, \Delta t_2, ..., \Delta t_n$, as to the revision $r_1, r_2, ..., r_n$. The data in a *live matrix* are recalculated between revisions. The state of data in such a structure ideally changes more frequently and faster than the time it takes to extract the data from that structure. The period Δt_n during which *live matrix* is changing its state is much smaller than the period of time t_e needed to extract and store a single revision r_i of the matrix, as this would be a constantly moving target (Fig. 3). The data in the matrix are only consistent within one revision and become obsolete between revisions; thus, timing is crucial. The whole *live matrix* structure or any extracted file segments represent an inconsistent revision r_{incons} and quickly become obsolete.

The matrix keeps multiple file segments, which reside in many locations of the matrix structure. These are encrypted and/or hashed using standard algorithms, e.g, AES256, 3DES, SHA-2, SHA-3, and located in the matrix at a certain index.

Taking into account the nature of the data to be hashed, i.e., the instructions for controlling the manufacturing machine, self-driving car, IoT or any other data, a special-purpose hash function can be designed. Matrix vectors that are not accommodated by useful data can be populated with fake data—random data that resemble the original file.

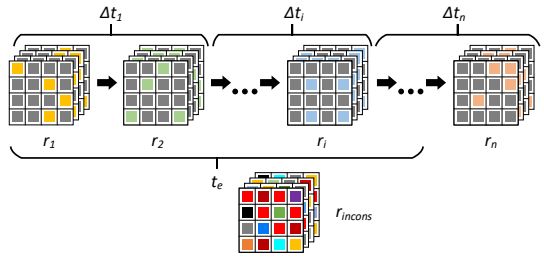


FIGURE 3. *Live matrix* state changes and inconsistent revision r_{incons}

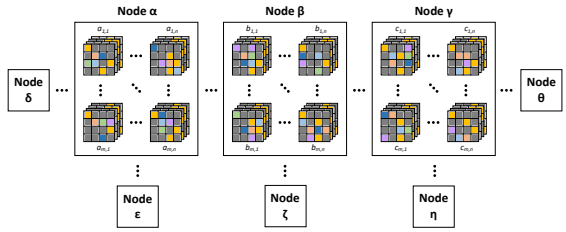


FIGURE 4. Secured distributed cloud storage

Distributed cloud storage can consist of one or multiple nodes α, β, γ, etc. Each node may consist of one or multiple multidimensional matrices a_{11} ... $a_{\text{m,n}}$, $b_{1,1}$ ... $b_{\text{m,n}}$, ..., etc. (e.g., Fig. 4). The density of each matrix can be set from 0% to 100%. For example, if the density is 10%, then only this ratio of values are filled in with the real segments of files. The rest of the values are synthetically generated data or information very similar to the actual data.

When the file is streamed from one location to another, the receiving location should also run a *live matrix* initialized with the matching encryption seed (based on time or other factors), and with a matching algorithmic setup. The stream comprises hashes of the file segment parts, $hints$. The actual information transferred in the stream is not the encrypted file parts, and there is no key to decrypt the streamed $hints$ (unless not constantly changing its state matrix is considered a key). As soon as actual bytes of the file are no longer transferred, there is no key as such, and there is no function to decrypt $hints$ (only to perform a look-up against the *live matrix* on the receiving end). We call this *key-less, byte-less information transfer* (Fig. 5).

When information is transferred to or from a non-cloud device (usually a data stream-consuming device with limited computing power, like a laptop, a manufacturing machine, a smart car, etc.) there is no physical possibility to keep more than a couple of revisions of live matrices on that side. It is thus impossible for such a device to decrypt the stream even thirty seconds later. This makes it impossible to "replay" the stream if an attacker records a fragment or even the whole stream.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

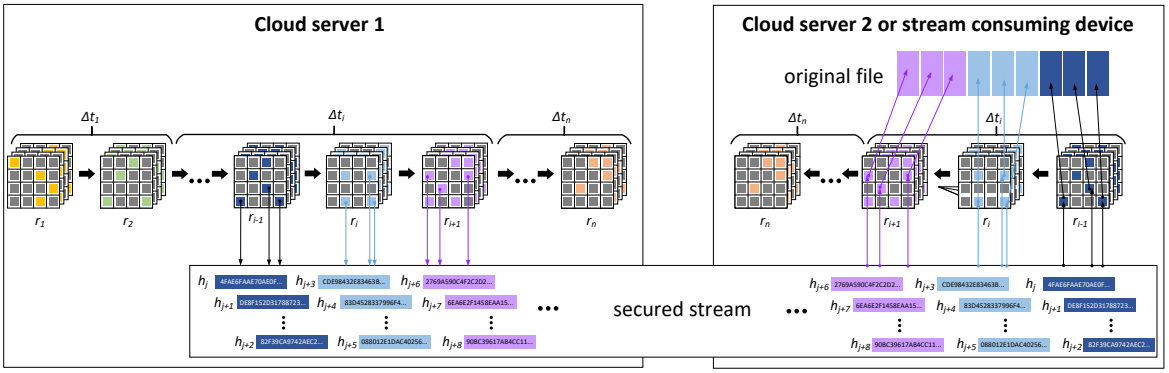


FIGURE 5. Key-less, byte-less secured streaming

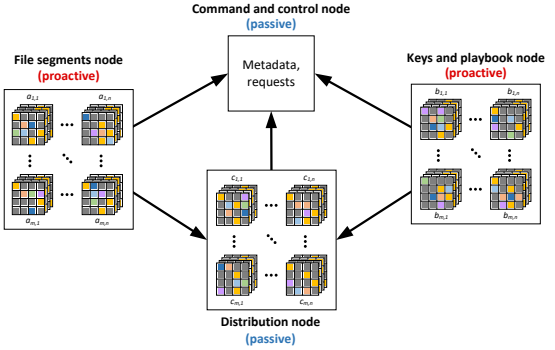


FIGURE 6. Types of nodes

D. PROACTIVE AND PASSIVE CLOUD NODES

A proactive cloud node is a cloud node or group of nodes that is autonomous to a certain extent. Proactive nodes do not expose any inbound TCP/UDP ports or APIs over a local or public network. Proactive nodes are the initiators of any communication between proactive and passive cloud nodes. Passive nodes cannot initiate the communication with proactive nodes; they need to wait for a request from one of the proactive nodes. Passive nodes only reply over the local network to requests incoming from proactive nodes.

Proactive nodes are used to store file segment data, users' public keys, file segments, encryption keys, and streaming playbooks. Passive nodes store metadata and run jobs, e.g., stream data outside the cloud, or deliver data at the right time at the right place, like a manufacturing machine or a self-driving car. The differentiation between proactive and passive cloud node types supports an important principle of segmentation in data security. Moreover, proactive cloud nodes rely on detective controls mechanisms [65] to analyze activity, events, logs, history of node communication, etc. Proactive cloud nodes can implement basic through the most sophisticated detective control methods using artificial intelligence, honeypotting, intrusion detection systems, etc. [66], [67].

E. PROTOCOL

A simplified protocol is presented in Fig. 7, 8, 9 and 10. All communication between cloud nodes is encrypted, although this is not explicitly shown in the figures for the sake of clarity. Fig. 7 describes file upload by the user and secure storage of that file in the cloud. Fig. 8 and 9 describe storage maintenance over time and *live matrix* recalculation, respectively. There are two options to achieve the recalculation of storage: Fig. 8 depicts the use of a newly created set of keys, while Fig. 9 depicts utilization of the homomorphic properties of encryption methods. In the latter case, each file segment is recalculated by performing a homomorphic operation on a file segment. Thus, no additional key generation and exchange is necessary. The secured streaming protocol is depicted in Fig. 10.

V. IMPLEMENTATION

The highly secure distributed file storage and transfer solution setup (Fig. 6) consists of four types of nodes:

a) The *command and control node* is responsible for storing command and control metadata. For example, whenever it is time to run a periodic job to re-encrypt file segments with a different set of keys, the file segments node and keys and playbook node communicate through this node

b) The *file segments node* keeps the file segments in live matrices, performs a scheduled or on-demand recalculation of hashes or re-encryption of file segments, analyzes the behavior of the command and control node and distribution nodes, and makes corresponding decisions, for example, to support a streaming session initiated by the distribution node. Moreover, this node has controls that measure the speed of data consumption and compare it with realistic consumption rates. If the rate at which data are requested or consumed by the distribution node is faster than expected, an alarm state is triggered for a certain streaming session, or perhaps

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

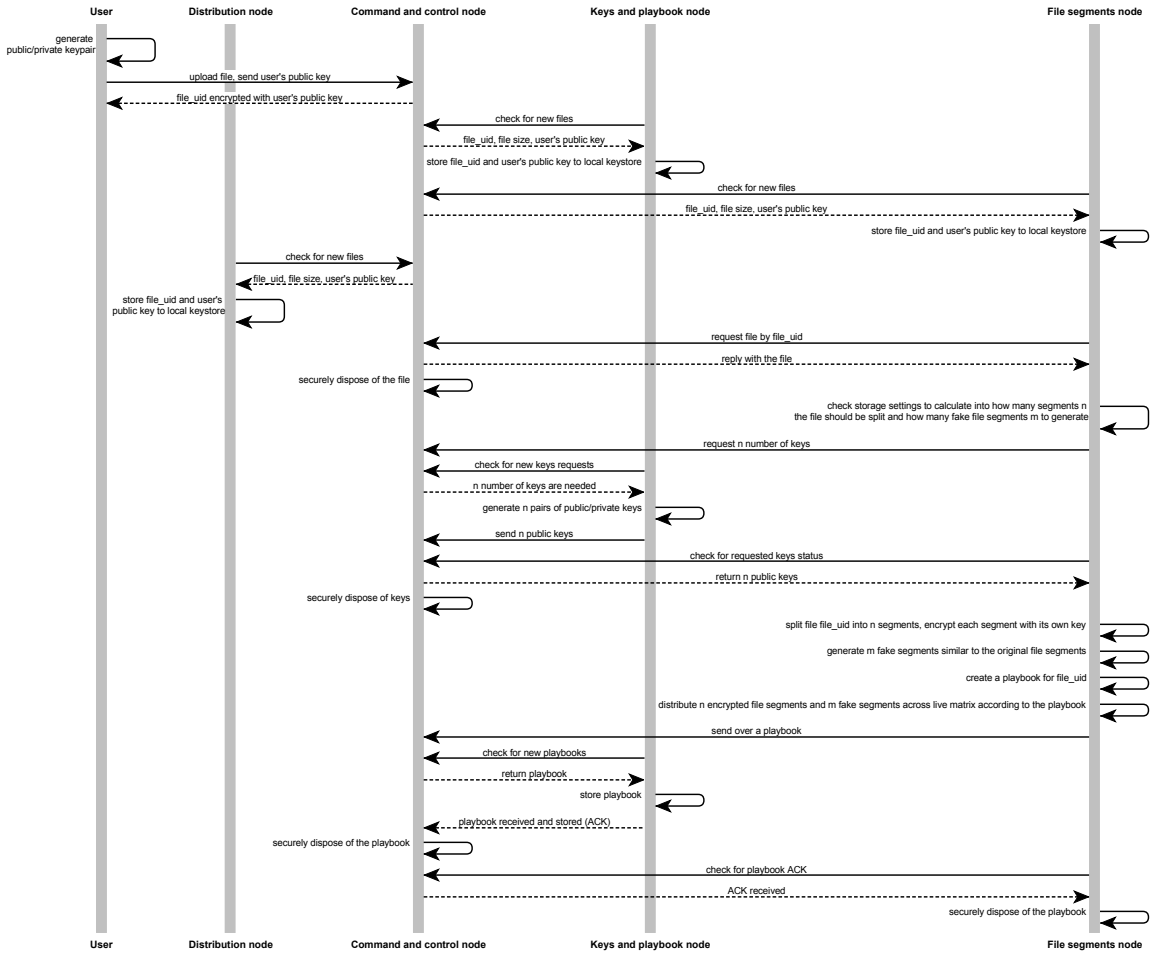


FIGURE 7. Protocol for storing a file in secured cloud storage

all sessions, depending on the setup and protection level desired

c) The *keys and playbook* node is responsible for secure storage of keys and playbooks. Playbooks describe the sequence of segments in the file segments node. Without the right key, it is impossible to decrypt the file segment; conversely, without the right playbook, it is virtually impossible to locate and extract the desired data from storage. Depending on the setup, the keys and playbook node can deliver the right keys at the right time to the right place (e.g., to an AM machine which has already received a secured stream from the distribution node has recreated the file segments from $hints$ using a locally running *live matrix*, and now needs to decrypt the data from file segments to produce the part). In the alarm state, this node stops the streaming process and stops issuing keys

d) The *distribution node* runs content distribution jobs to transfer files to external sources, like other secured clouds or AM machines or self-driving cars. This node isolates different streaming jobs, optimizes streaming speed based on data transfer rate, and performs data delivery checks in the stream. It also participates in the authorization scheme for external cloud and stream consumption devices.

The setup can be extended, so each node type is a sub-cloud of multiple machines implementing distributed live matrices (Fig. 4).

VI. SECURITY EVALUATION

Our threat modeling and security analysis is based on several well-defined threat frameworks from Behl et al. [68]–[70] and Saripalli et al. [71]. The latter provides the list of "Threat events compromising cloud security" [71], which our dis-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

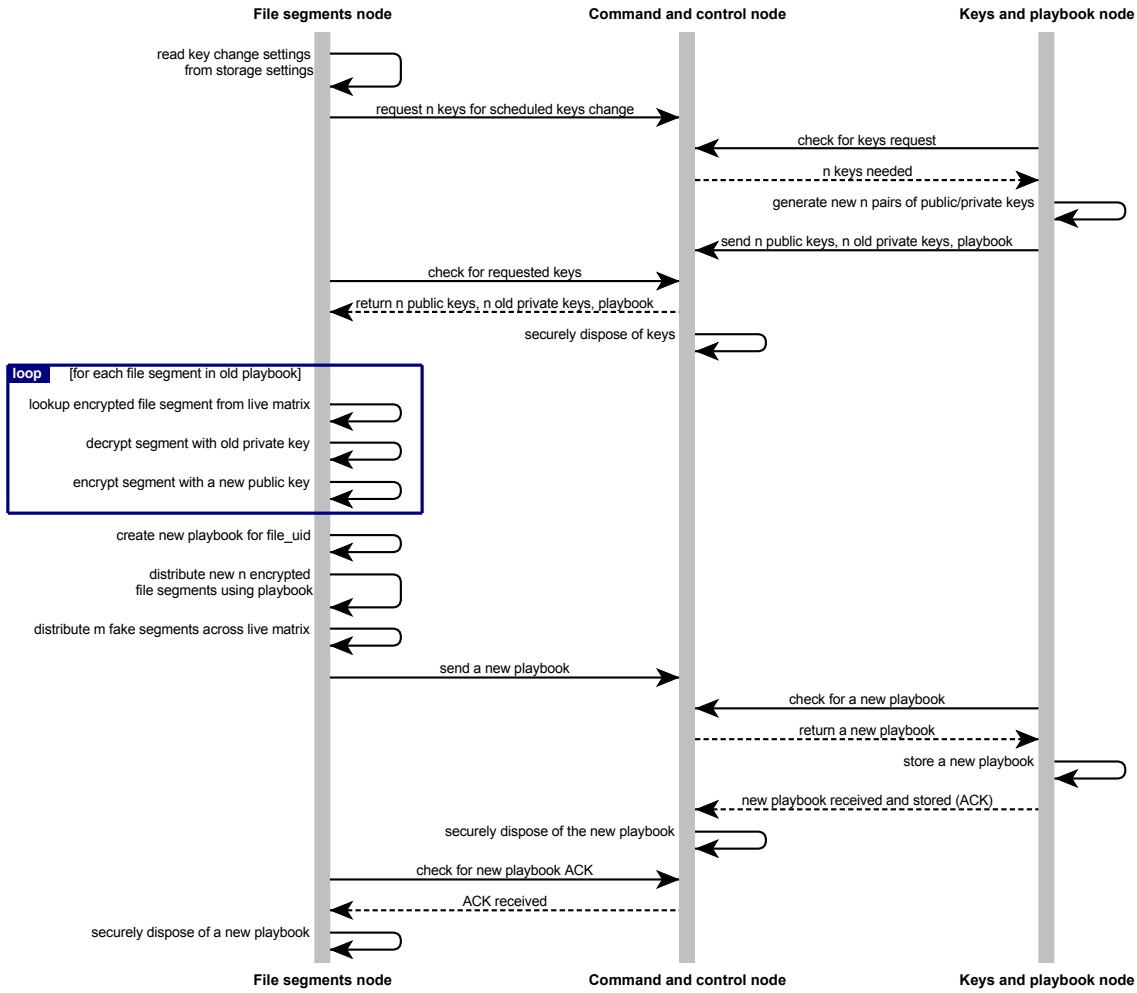


FIGURE 8. Protocol for storage maintenance and *live matrix* recalculation with key change

tributed storage and transfer solution is intended to address. These are: a) *Isolation failure*: failure to effectively separate storage, memory, and routing causes isolation failure; b) Malicious insider at cloud provider: a cloud provider's employee maliciously alters or corrupts customer data; c) *Intercepting data in transit*: failure in cryptographic techniques leads to data sniffing, spoofing and man-in-the-middle attacks during transit; d) *Data Leakage on Up/Down*: interception of data between the customer and the cloud provider leads to leakage of data to third parties; e) *Loss of encryption keys*: exposure of customer's secret keys to malicious parties.

We have evaluated the relevant threats and created a threat model, summarized, along with the corresponding mitigation, in Table 1. We have derived the most important attack vectors from our threat model and provide an analysis.

If an attacker is able to pose as an authorized user, he still cannot download the data unless he digitally signs and submits a transaction to stream data to a data consumer.

There is no single point of compromise. If an attacker is able to access one of the node types, he still won't be able to extract data. An attacker needs to get access to at least two different types of node to decrypt the data. File segment nodes, keys nodes, and playbook nodes are proactive and do not expose any TCP ports. Thus, there is no way for the attacker to log in to these nodes unless they get access to a virtual machine or physical hardware and scan memory to get the contents from the running application.

If an attacker is able to obtain a playbook file, it is still only one instance. The attacker will not have access to every modified instance of the playbook. Without continuous

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

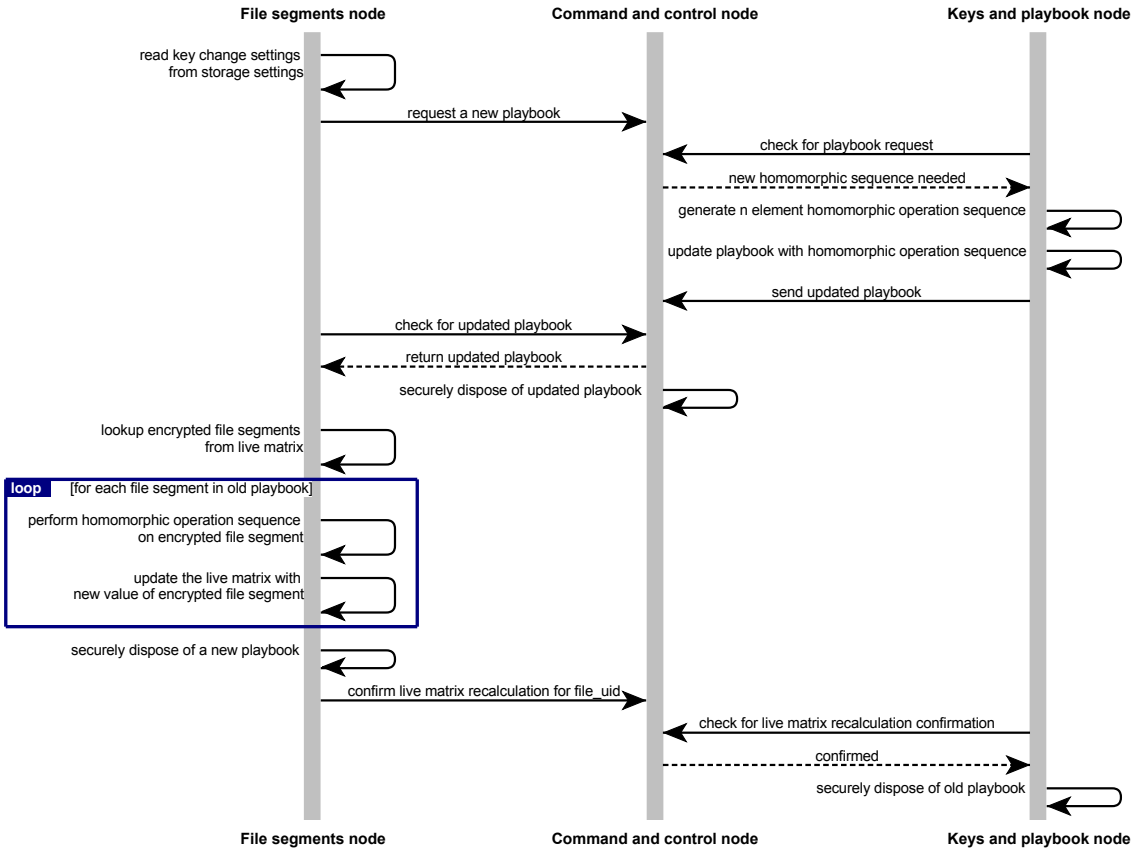


FIGURE 9. Protocol for storage maintenance and *live matrix* recalculation with homomorphic encryption

updates, the attacker will not have access to the data.

Even if an attacker captures the data stream during a streaming session, it rapidly becomes obsolete very soon, unless the attacker obtained a seed to start and run *live matrix*. During a single streaming session, the data are from different *live matrix* revisions, so in order to decrypt the data, corresponding *live matrix* states should be obtained. This can be done by compromising the server side or the data consumer side. This is still difficult; for instance, if the attacker gains access to the data consumer side, then he needs to be present from the very beginning of the stream and record the low-level machine code as it is transmitted. The solution depends on the exact data consumer implementation. For example, in the case of holographic video streaming, 3D printing, and other types of AM, data that are already consumed must be disposed of just after consumption. Additionally, if the attacker obtains one full unencrypted sequence of machine code, then this sequence could be used on exact make and model of the manufacturing machine, which makes it harder to distribute and violate the copyright.

In our previous research [5], [33], [34], we assumed that data—once taken from some kind of secured storage—are decrypted and then encrypted with a different method for delivery to the data consumer. Then, the distribution node can also be a point of attack. In that case, the attacker can obtain a file or a stream on the server during re-encryption between storage and streaming. However, in the current approach, there is no need for transcoding the data.

In TLS file transfer, a certain key is used to encrypt data; if an attacker obtains the key, he can decrypt the file. Such keys are often reused by the services, or changed infrequently, making them vulnerable to collision and brute-forcing over time. In our solution, the complexity to brute-force the keys increases exponentially: the file is split into thousands of segments and the *live matrix* is constantly recalculated.

If an attacker obtains access to the data consumption device, he can receive file parts over a long period of time. There is no way to get all the files from the storage. Consequently, during one session, only one file can be obtained, and over a comparably long period of time. For example,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

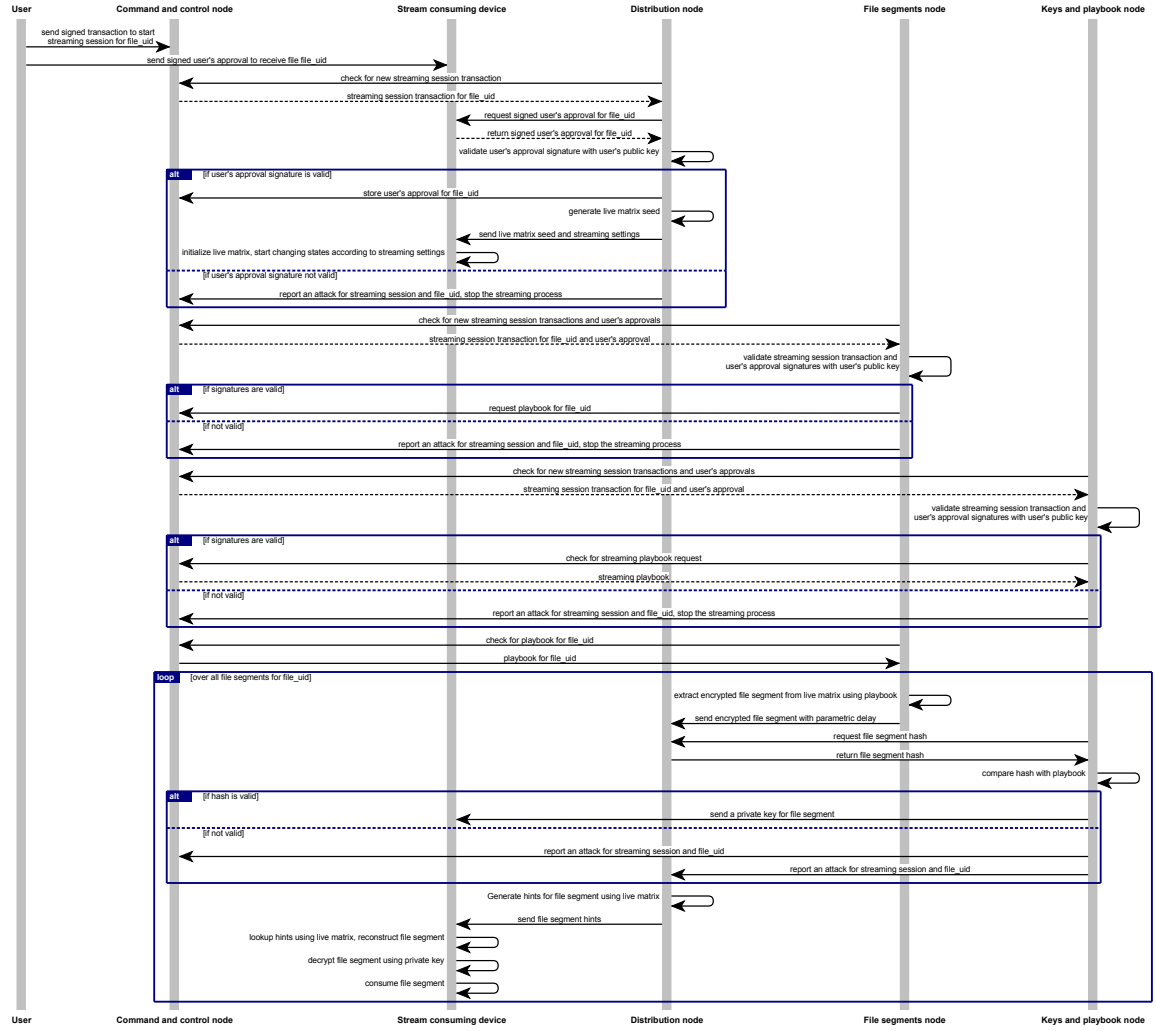


FIGURE 10. Protocol for secured streaming

producing a part using automated manufacturing can take days, and movies can last for hours. For an attacker acting this way, it would be inefficient time-wise to extract data from secured storage; this would not allow getting all the data from the secured cloud storage.

If an attacker starts to request more file parts within a shorter time-frame than a certain threshold, then the distribution node stops providing data. If such an attack is performed on a consuming device or a data channel, stopping the stream makes it impossible to get the rest of the file.

If an attacker carries out an attack on the secured cloud or a stream to a data consumer, secure cloud nodes collect the data and compare the metrics with those in configuration files. All abnormal activities, events, and logs can trigger an alarm state. A hacker would need to carry out a comprehensive analysis for a considerable period to figure out which changes in communication would cause an alarm state. By that time, the hacker would likely be detected, and mitigation procedures executed.

VII. PERFORMANCE EVALUATION

In our lab-level implementation, we used the distributed database Cassandra [72] to implement live matrices, the Apache Spark near real-time distributed scale data processing [73] with the Java programming language to implement operations over matrices, and Apache Kafka [74] to maintain a queue of service requests and streaming jobs.

We stored file segments in column families of four bytes each, encrypted with public keys, in Cassandra. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

TABLE 1. Threat model

Threat	Mitigation
Cloud multi-tenancy	The end user does not have direct access to the data; he can only issue a start streaming command signed with his private key. There is no way to co-locate new malicious data with a victim's assets.
Elasticity	File segments are encrypted at rest and shuffled across the matrix; without a valid playbook, there is no way to extract the data. Moreover, after scheduled maintenance and recalculation of *live matrix*, the old playbook is discarded and a new playbook generated, so old data quickly become obsolete.
Availability of information	Although this particular method and its implementation does not currently provide comprehensive data backup options, it is intended for secured data distribution. The data owner should store a copy on offline media.
Cloud management layer	The solution is intended to run on cloud infrastructure such as AWS, Azure, or GCE, and this threat is mitigated by the cloud provider.
Information integrity and privacy	All transactions by the user to delete or transfer data from the secured cloud should be signed by a private key; then, nodes will independently verify the digital signature with the public key of the user. If the signature is not valid, no transfer is performed.
Cloud secure federation	The user uses a public-private key-pair to manage his resources at different locations. To perform streaming of the file to the data consumer, the user must sign a streaming transaction and send it to the secured cloud. A user can sign another transaction and send it to the data consumer, so the consumer can prove his eligibility to receive a secured data stream.

public/private key generation, we used elliptic curve type secp256k1 [75]. We used the last 20 bytes of the public key to uniquely index encrypted file segments in the Cassandra column families. We used Apache Spark to re-encrypt the file segments and recalculate new indexes in Cassandra every two minutes, then provided the updated version of the playbook to the keys and playbook node. For hashing, we used the Keccak-256 [37] hash function.

We used a local cloud of four bare-metal physical machines to run the software. Two machines (one for the file segments node and one for the distribution node) each had an 8x GPU AMD Radeon RX580 chipset with 8 GB GDDR5, i7 CPU, 16 GB RAM, and 128 GB SSD. The other two machines, used for the command and control node and the keys and playbook node, respectively, had an Intel Celeron processor, 4 GB RAM, and 32 GB SSD. We set the GPUs in a computing mode and flashed them with a modified firmware for higher hash rates. On average, each GPU was able to produce 31.5 Mhash/s; a few outstanding GPUs performed at 28.5 Mhash/s. We achieved an average hash-rate of 248 Mhash/s total on each of the machines equipped with 8x GPUs.

First, we tested secured streaming between two cloud machines. We performed secured streaming of the 20 MB file in the local network from the file segment node to the distribution node. We were able to recalculate hashes for the three-dimensional *live matrix* of 256^3 at an average of 14 revisions per second. In another test, we were able to recalculate a bigger matrix of 4096^3 with an average of one revision every 5 minutes. Second, we carried out the test between a cloud node and a data consumer node. For this test, we needed one more machine. The external stream receiving side was a laptop with an i7 processor, 8 GB RAM, and 256 GB SSD, GPU AMD Radeon RX570 chipset with 2 GB RAM, intended to emulate a single user consuming the stream. The GPU of this machine was able to produce 18 MHash/s. For this test, we needed to use only one GPU on the distribution node, with timing matching the calculation speed of the receiving machine. We were able to recalculate hashes for the three-dimensional *live matrix* of 256^3 states at an average rate of one per second. In another test, we performed a streaming session on the bigger *live matrix* of 4096^3, and we were able to calculate a new state on average every 64 minutes. The results are reflected in Table 2.

TABLE 2. Proposed method performance (base rates 100 Mhash per h = \$0.05, 1 GB = \$0.087) for a 20 MB file size

Experiment	Matrix size	State change frequency	Δ traffic (GB)	Δ cost (\$)
cloud-to-cloud	256^3	14 per s	1.67	0.1457
cloud-to-cloud	4096^3	1 per 5 min	0.85	0.0751
cloud-to-stream consuming device	256^3	1 per s	1.67	0.1437
cloud-to-stream consuming device	4096^3	1 per 64 min	0.85	0.0711

We performed additional tests with different file sizes: 41 MB, 119 MB, 583 MB, and 1.1 GB. The results showed a linear dependency for overhead traffic and overhead server costs. In future research, we will seek to reduce overhead traffic.

The tests showed that overhead increases with smaller matrix sizes. This is a result of change in the matrix size to hash function output ratio in bytes. We recommend the use of proven SHA cryptographic functions, even if this creates a bigger overhead. If minimizing bandwidth is important, then hash functions with a smaller length output should be selected.

We also confirmed that the cloud can adapt to the computing power capacity available on the consumer's end and produce a stream that could be consumed with less computing power. With the increase in computing power needed to calculate *live matrix* revisions, the power needed by a hacker to try to decode the stream would increase exponentially. Our results show that catching such a stream would be an ever-moving target. Even in the case of success, the information would become obsolete very quickly, making it hard to carry out any analysis to decrypt the file being transferred.

VIII. CONCLUSION

In this paper, we described and evaluated an approach that leverages the physical limitations of the computational process into a defense strategy to make cloud file storage and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946730, IEEE Access

transfer highly secure. The method was designed to fulfill multiple important requirements for the use cases we discussed. The data transfer is lossless, so this method will work not only for delivering machine code to manufacturing machines, but for many other applications, including audio and video streaming. The most notable features of our approach are: a) The solution is tightly coupled with secured storage, so there is no need for re-encryption in order to stream to remote data consumers, like other clouds or AM machines; b) By its nature, this solution keeps the data in partitions, and streaming also implies partition tolerance on file transfer. The data are segmented, and there are security controls based on the physical limitations of the computational process—it is not physically possible to extract and consume all the data within a reasonable time-frame; c) If multiple machines are used for each type of node, then there is no single point of failure in case of intrusion or fault; d) It may be used for peer-to-peer information transfer, though this requires the *live matrix* engine be installed on the peer machines participating in the information transfer; e) The solution can send bi-directional streams; on the receiving side, *live matrix* could be used for the reverse stream, for example, to transmit telemetry or interactive feedback.

In future work, we will concentrate on data storage fault tolerance mechanisms and intelligent adaptiveness to available computing power and network bandwidth.

Acknowledgement

This research was partially supported by Astra6-1 project[Project-code: 2014-2020.4/01.16-0032].

REFERENCES

[1] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Infocom, 2010 proceedings IEEE. Ieee, 2010, pp. 1–9.

[2] "3dprinteros cloud world statistics," https://cloud.3dprinteros.com/dashboard/#world-statistics, accessed: 31-05-2019.

[3] M. Molitch-Hou, "Dremel and 3dprinteros partner for 3d printing in the cloud," Engineering.com, Jan 2017. [Online]. Available: https://www.engineering.com/3DPrinting/3DPrintingArticles/ArticleID/14021/Dremel-and-3DPrinterOS-Partner-for-3D-Printing-in-the-Cloud.aspx

[4] S. Saunders, "Thanks to new partnership, 3dprinteros software will now power kodak portrait 3d printers," 3DPrint.com, May 2018. [Online]. Available: https://3dprint.com/214798/3dprinteros-kodak-portrait/

[5] K. Isbjörnssund and A. Vedeshin, "Secure streaming method in a numerically controlled manufacturing system, and a secure numerically controlled manufacturing system," Dec. 3 2015, uS Patent App. 14/761,588.

[6] R. Landauer, "Fundamental physical limitations of the computational process," Annals of the New York Academy of Sciences, vol. 426, no. 1, pp. 161–170, 1984.

[7] S. Kalpakjian, Manufacturing engineering and technology. Pearson Education India, 2001.

[8] "Slm280 2.0," https://www.slm-solutions.com/en/products/machines/slmr280-20/, accessed: 15-07-2019.

[9] "Stratasys objet1000 plus," https://www.stratasys.com/3d-printers/objet1000-plus, accessed: 15-07-2019.

[10] "Materialise magics," https://www.materialise.com/en/software/magics, accessed: 15-07-2019.

[11] C. Okwudire, S. Huggi, S. Supe, C. Huang, and B. Zeng, "Low-level control of 3d printers from the cloud: A step toward 3d printer control as a service," Inventions, vol. 3, no. 3, p. 56, 2018.

[12] "Prusa stepper motor calculator. steps per millimeter - belt driven systems," https://blog.prusaprinters.org/calculator/, accessed: 16-07-2019.

[13] "Achievable step rates," https://reprap.org/wiki/Step_rates, accessed: 16-07-2019.

[14] "Klipper 3d," https://www.klipper3d.org/Features.html, accessed: 16-07-2019.

[15] Y. Nilsiam and J. M. Pearce, "Free and open source 3-d model customizer for websites to democratize design with openscad," Designs, vol. 1, no. 1, p. 5, 2017.

[16] M. Kintel and C. Wolf, "Openscad," GNU General Public License, p GNU General Public License, 2014.

[17] A.-M. Nergi, "Eestis pusti pandud innovatsiooniakadeemia liigub edasi euroopasse," Arileht.ee, May 2013. [Online]. Available: https://arileht.delfi.ee/news/uudised/eestis-pusti-pandud-innovatsiooniakadeemia-liigub-edasi-euroopasse?id=67036706

[18] "Telerik fiddler," https://www.telerik.com/fiddler, accessed: 17-07-2019.

[19] G. Brunette, R. Mogull et al., "Security guidance for critical areas of focus in cloud computing v2. 1," Cloud Security Alliance, pp. 1–76, 2009.

[20] S. Mastorakis, "Peer-to-peer data sharing in named data networking," Ph.D. dissertation, UCLA, 2019.

[21] S. Mastorakis, A. Afanasyev, Y. Yu, and L. Zhang, "ntorrent: Peer-to-peer file sharing in named data networking," in 2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2017, pp. 1–10.

[22] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang et al., "Named data networking," ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66–73, 2014.

[23] M. G. Jaatun, G. Zhao, and S. Alapnes, "A cryptographic protocol for communication in a redundant array of independent net-storages," in Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on. IEEE, 2011, pp. 172–179.

[24] E. Miller, D. Long, W. Freeman, and B. Reed, "Strong security for distributed file systems," in Performance, Computing, and Communications, 2001. IEEE International Conference on. IEEE, 2001, pp. 34–40.

[25] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1–30, 2006.

[26] C. Rong and W.-C. Kim, "Effective storage security in incompletely trusted environment," in null. IEEE, 2007, pp. 432–437.

[27] V. Manasa and M. Vikram, "A secured adaptive mobile video streaming and efficient social video sharing in the clouds," International Journal of Computer Science and Information Technologies(IJCSIT), vol. 5, no. 4, pp. 5153–5156, 2014.

[28] M. Bucicoiu, M. Ghideu, and N. Tăpus, "Secure cloud video streaming using tokens," in RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, 2014. IEEE, 2014, pp. 1–6.

[29] Z. Chen, H. Yin, C. Lin, and L. Ai, "3d-wavelet based secure and scalable media streaming in a centralcontrolled p2p framework," in Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on. IEEE, 2007, pp. 708–715.

[30] S.-H. Liu, H.-Y. Yu, J.-Y. Wu, J.-J. Chen, J.-L. Liu, and D.-H. Shiue, "A secured video streaming system," in System Science and Engineering (ICSSE), 2010 International Conference on. IEEE, 2010, pp. 625–630.

[31] S. J. Wee and J. G. Apostolopoulos, "Secure scalable video streaming for wireless networks," in Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on, vol. 4. IEEE, 2001, pp. 2049–2052.

[32] N.-H. Lin, T.-H. Huang, and B.-Y. Chen, "3d model streaming based on jpeg 2000," IEEE Transactions on Consumer Electronics, vol. 53, no. 1, 2007.

[33] P.-M. Sepp, A. Vedeshin, and P. Dutt, "Intellectual property protection of 3d printing using secured streaming," in The Future of Law and eTechnologies. Springer, 2016, pp. 81–109.

[34] K. Isbjornssund and A. Vedeshin, "Method and system for enforcing 3d restricted rights in a rapid manufacturing and prototyping environment," Feb. 27 2014, uS Patent App. 13/973,816.

[35] E. W. Tischhauser, "Mathematical aspects of symmetric-key cryptography," Ph.D. dissertation, Ph. D. thesis, Katholieke Universiteit Leuven, 2012.

[36] V. G. Cerf, "Ascii format for network interchange," 1969.

[37] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Keccak sponge function family main document," Submission to NIST (Round 2), vol. 3, no. 30, 2009.

[38] B. Schneier, "Attack trees," Dr. Dobb's journal, vol. 24, no. 12, pp. 21–29, 1999.
[39] A. Biryukov, J. Lano, and B. Preneel, "Cryptanalysis of the alleged securid hash function," in International Workshop on Selected Areas in Cryptography. Springer, 2003, pp. 130–144.
[40] J. A. Jacobs, Geomagnetic micropulsations. Springer Science & Business Media, 2012, vol. 1.
[41] R. Fowler, B. Kotick, and R. Elliott, "Polarization analysis of natural and artificially induced geomagnetic micropulsations," Journal of Geophysical Research, vol. 72, no. 11, pp. 2871–2883, 1967.
[42] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams, "Quantum clock synchronization based on shared prior entanglement," Physical Review Letters, vol. 85, no. 9, p. 2010, 2000.
[43] M. Xu, D. A. Tieri, E. Fine, J. K. Thompson, and M. J. Holland, "Synchronization of two ensembles of atoms," Physical review letters, vol. 113, no. 15, p. 154101, 2014.
[44] C. De Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles," in International Conference on Information Security. Springer, 2006, pp. 171–186.
[45] P. Ekdahl and T. Johansson, "A new version of the stream cipher snow," in International Workshop on Selected Areas in Cryptography. Springer, 2002, pp. 47–61.
[46] J. D. Golić, "Cryptanalysis of alleged a5 stream cipher," in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 1997, pp. 239–255.
[47] M. Hell, T. Johansson, and W. Meier, "Grain: a stream cipher for constrained environments," IJWMC, vol. 2, no. 1, pp. 86–93, 2007.
[48] R. Anderson and C. Manifavas, "Chameleon-a new kind of stream cipher," in International Workshop on Fast Software Encryption. Springer, 1997, pp. 107–113.
[49] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius, "Rabbit: A new high-performance stream cipher," in International Workshop on Fast Software Encryption. Springer, 2003, pp. 307–329.
[50] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier et al., "Sosemanuk, a fast software-oriented stream cipher," in New stream cipher designs. Springer, 2008, pp. 98–118.
[51] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in International workshop on cryptographic hardware and embedded systems. Springer, 2007, pp. 450–466.
[52] J. Daemen, L. Knudsen, and V. Rijmen, "The block cipher square," in International Workshop on Fast Software Encryption. Springer, 1997, pp. 149–165.
[53] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The led block cipher," in International Workshop on Cryptographic Hardware and Embedded Systems. Springer, 2011, pp. 326–341.
[54] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," in International Workshop on Fast Software Encryption. Springer, 1993, pp. 191–204.
[55] R. A. Rueppel, "Stream ciphers," in Analysis and Design of Stream Ciphers. Springer, 1986, pp. 5–16.
[56] P. Mahajan and A. Sachdeva, "A study of encryption algorithms aes, des and rsa for security," Global Journal of Computer Science and Technology, 2013.
[57] P. Trakadas, T. Zahariadis, H. Leligou, S. Voliotis, and K. Papadopoulos, "Analyzing energy and time overhead of security mechanisms in wireless sensor networks," in 2008 15th International Conference on Systems, Signals and Image Processing. IEEE, 2008, pp. 137–140.
[58] C. Xenakis, N. Laoutaris, L. Merakos, and I. Stavrakakis, "A generic characterization of the overheads imposed by ipsec and associated cryptographic algorithms," Computer Networks, vol. 50, no. 17, pp. 3225–3241, 2006.
[59] P.-L. Cayrel, G. Hoffmann, and M. Schneider, "Gpu implementation of the keccak hash function family," in International Conference on Information Security and Assurance. Springer, 2011, pp. 33–42.
[60] L. Dadda, M. Macchetti, and J. Owen, "The design of a high speed asic unit for the hash function sha-256 (384, 512)," in Proceedings of the conference on Design, automation and test in Europe-Volume 3. IEEE Computer Society, 2004, p. 30070.
[61] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 3–16.
[62] J. R. Goodman, "Using cache memory to reduce processor-memory traffic," ACM SIGARCH Computer Architecture News, vol. 11, no. 3, pp. 124–131, 1983.
[63] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of internet services and applications, vol. 1, no. 1, pp. 7–18, 2010.
[64] D. P. Bovet and M. Cesati, Understanding the Linux Kernel: from I/O ports to process management. " O'Reilly Media, Inc.", 2005.
[65] R. K. Ko, B. S. Lee, and S. Pearson, "Towards achieving accountability, auditability and trust in cloud computing," in International Conference on Advances in Computing and Communications. Springer, 2011, pp. 432–444.
[66] B. Nagpal, N. Singh, N. Chauhan, and P. Sharma, "Catch: Comparison and analysis of tools covering honeypots," in Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in. IEEE, 2015, pp. 783–786.
[67] A. S. Sohal, R. Sandhu, S. K. Sood, and V. Chang, "A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments," Computers & Security, vol. 74, pp. 340–354, 2018.
[68] A. Behl and K. Behl, "An analysis of cloud computing security issues," in Information and Communication Technologies (WICT), 2012 World Congress on. IEEE, 2012, pp. 109–114.
[69] ——, "Security paradigms for cloud computing," in Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on. IEEE, 2012, pp. 200–205.
[70] A. Behl, "Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation," in Information and communication technologies (WICT), 2011 world congress on. IEEE, 2011, pp. 217–222.
[71] P. Saripalli and B. Walters, "Quirc: A quantitative impact and risk assessment framework for cloud security," in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, 2010, pp. 280–288.
[72] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," ACM SIGOPS Operating Systems Review, vol. 44, no. 2, pp. 35–40, 2010.
[73] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin et al., "Apache spark: a unified engine for big data processing," Communications of the ACM, vol. 59, no. 11, pp. 56–65, 2016.
[74] N. Garg, Apache Kafka. Packt Publishing Ltd, 2013.
[75] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in International Conference on Financial Cryptography and Data Security. Springer, 2014, pp. 157–175.

Appendix 3

III

A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. Smart cyber-physical system for pattern recognition of illegal 3d designs in 3d printing. In *Communications in Computer and Information Science*, pages 74–85. Springer International Publishing, 2020

Smart Cyber-Physical System for Pattern Recognition of Illegal 3D designs in 3D Printing

Anton Vedeshin[1], John Mehmet Ulgar Dogru[1],
Innar Liiv[2], Sadok Ben Yahia[2], and Dirk Draheim[3]

[1] 3DPrinterOS, 3D Control Systems, Inc., San Francisco, CA, USA
{anton,john}@3dprinteros.com
https://3DPrinterOS.com
[2] Department of Software Science, Tallinn University of Technology, Tallinn, Estonia
{innar,sadok.ben}@taltech.ee
[3] Information Systems Group, Tallinn University of Technology, Tallinn, Estonia
dirk.draheim@taltech.ee

Abstract. The method to protect intellectual property (IP) in automated manufacturing (AM) and 3D printing industry particularly, presented in this paper, is based on a smart cyber-physical system and the radical improvement of preventive and detective controls to find potential cases of automated manufacturing copyrights infringement. The focus of this paper is not the ecosystem of managing a large network of physical 3D printers, but a smart application and data analysis of data flow within the ecosystem to solve a problem of IP protection and illegal physical objects manufacturing. In this paper, we focus on the first step in this direction – pattern recognition of illegal physical designs in 3D printing, and detection of firearms parts particularly. The proposed method relies on several important steps: normalization of 3D designs, metadata calculation, defining typical illegal designs, pattern matrix creation, new 3D designs challenging, and pattern matrix update. We classify 3D designs into loose groups without strict differentiation, forming a pattern matrix. We use conformity and seriation to calculate the pattern matrix. Then, we perform the analysis of the matrix to find illegal 3D designs. Our method ensures simultaneous pattern discovery at several information levels - from local patterns to global. We performed experiments with 5831 3D designs, extracting 3728 features. It took 12 minutes to perform pattern matrix calculation based on the test data. Each new 3D design file pattern recognition took 0.32s on four core, 8GB ram, 32GB SSD Azure VM instance.

Keywords: Pattern recognition · intelligent manufacturing systems · technology social factors · distributed computing

1 Introduction

The method to protect intellectual property (IP) in automated manufacturing (AM) and 3D printing industry particularly, presented in this paper, is based

on the radical improvement of preventive and detective controls on software, firmware, and hardware levels of AM machines. These controls would help to find potential cases of copyright infringement, illegal objects, and firearms manufacturing with the use of AM machines. In this paper, we focus on the first step in this direction – pattern recognition of illegal physical objects in 3D printing, and detection of 3D designs with firearm parts.

Recently, we have witnessed the advent of cloud manufacturing, where F500 enterprises, small and medium businesses, and home users use devices such as 3D printers, CNC mills, laser jets, and robotics to manufacture products locally at the point and time of need. The impressively fast adoption of these technologies strongly indicates that this novel approach to manufacturing can become a crucial enabler for the real-time economy of the future, i.e., a possible paradigm shift in manufacturing towards cloud manufacturing. Now it is possible to manufacture a real working part or a usable product from a CAD design in just hours using cloud manufacturing. Companies and people would not buy a ready-made product at the shop, but obtain raw material and produce products locally, utilizing their own or nearby accessible automated manufacturing machinery. With all the benefits of the new way to manufacture things, there is a growing threat to society. Firstly, there is a need to protect intellectual property (IP) in the form of 3D designs and manufacturing files. Secondly, there is an increased risk and real cases [5, 17, 33] of people producing firearms at home using desktop 3D printers.

With the growing popularity of automated manufacturing (AM), robotic process automation (RPA), the need to protect the intellectual property (IP) at every stage of the AM process became more important: from idea, CAD design to machine instructions and manufacturing files. Companies and people should be able to protect their IP by claiming their technical, mechanical, and chemical solutions through a decentralized platform that protects their IP. Moreover, there should be a measure established which would protect the society from a potential leakage of firearm designs and manufacturing of illegal objects.

In this paper, we are going to address this problem and present one of the possible solutions using a smart cyber-physical system. Our proposed digital ecosystem for personal manufacturing enables one to link the physical world (3D printers) with virtual cloud-based operating system [29]. The focus of this paper is not the ecosystem of managing a large network of physical 3D printers, but a smart application and data analysis of data flow within the ecosystem to solve a problem of IP protection and illegal physical objects manufacturing. In the following, we glance at the main contributions of this paper:

1. We discuss the motivation for the creation of the cloud-based manufacturing operating system to address an evolving critical problem of IP protection and illegal physical objects manufacturing;
2. We introduce a novel cloud-based manufacturing operating system architecture to protect IP and detect illegal physical objects using pattern recognition;

The remainder of the paper is organized as follows: In Section 2, we discuss the related work and our motivation to do this task. In Section 3, we describe the logic behind patter recognition for illegal 3D objects detection. In Section 4, we discuss the experiments and performance of our solution. In Section 5, we conclude the paper and give directions for future work.

2 Related work

In this section, we present related work and the overview of existing ways to protect IP copyright for automated manufacturing (AM).

In their work [13], Hou et al. cover traditional ways to secure 3D files. The authors describe solutions from digital rights management (DRM) to an embedding visual shapes into 3D printed models' internal structure. Later, the scanning of internal structures allows one to figure out it is an original part or a copy. Their work does not cover the whole AM workflow and focuses on protection methods, which can help after manufacturing already happened, for example, watermarking and tagging an object with RFID chips. Although it helps to detect whether the part was original, it does not protect from copying the part or preventing from manufacturing an illegal part. Moreover, by scanning a 3D printed part, it is hard to reverse engineer it due to internal structures, and the exact way it is manufactured. To reproduce the physical part, it is not enough to only obtain the shape of the object, 3D printer toolhead movements, speed, temperature used at that exact path - everything is important and affects the final physical properties of the object. IP protection is essential, mostly before manufacturing.

Nein-Hsien et al., in their research [23], describe a method to encode 3D models into a Jpeg stream, to transfer 3D designs. It is not a comprehensive solution and has definite limitations. Their paper does not handle AM end-to-end workflow, nor prevents an illegal 3D design manufacturing.

In our prior research [25] we only theoretically touched the protection of IP rights for 3D printing. We have described a paradigm applied to secure 3D content delivery called the *live matrix*. This prior work is purely theoretical, lacking technical details. The current paper is the first paper to explain IP copyright protection and illegal 3D part detection technically. This paper's contribution is to extend the initial idea with the implementation details, bring it to the next level, technically broaden it to the illegal 3D parts detection before the part is manufactured.

In our previous works [14, 15, 30], we have emphasized in detail the necessity to enforce the 3D files' copyrights through secured content delivery - 3D files streaming. Our previous work targets a very niche case to secure 3D designs data at rest and to be transferred from the server to a 3D printer. Previous solution [14, 15] is technically dense, has multiple drawbacks. Moreover, it lacks the protection of IP at any other stage of IP handling, nor illegal 3D part detection and manufacturing prevention of illegal parts [30]. In this paper, we contribute

a reliable and fast way to detect IP copyrights infringement and extend it to the illegal physical objects detection and prevention from manufacturing.

In their work [24], Mattingly et al. describe a method for three-dimensional printing with blockchain controls. Their work does not precisely describe how the copyright is given and what happens with the file. The idea of just closing the block with the list of transactions is a nature of blockchain; however, this is not clear, how it helps to grant access to the IP. The solution they propose is, basically, a log file that contains the records of what has happened in the past, regardless of whether the IP owner authorized the transaction or not. Smart contracts are not mentioned; thus, it is not clear how exactly the access is granted. The work lacks the details of implementation.

In their work [12], Holland et al. propose copyrights protection for additive manufacturing with a blockchain approach. They utilize the property of blockchain to eliminate the 3rd party so that the blockchain poses a trusted 3rd party or notary, which governs the exchange of transaction data. In their work, 3D design files are not stored in the blockchain, and the blockchain is used only for granting the license and number of prints. Their method lacks the process for the 3D printer to report to the blockchain. 3D printer should send a transaction to update how many times the file was printed and decrease the license quantity.

In the work [16], the authors present a method, which adds a distinctive nanomaterial chemical signature to the parts and registers it in the blockchain. Their method helps to check after the file was manufactured, whether it is an original file or a counterfeit. Their solution does not defend the file nor detects and prevents an illegal 3D file from being manufactured. In their work, the blockchain is a shared resource between the manufacturer of a 3D part and a part receiving party so that both have access to the blockchain instance, and then the counter-party can see the transactions made by the manufacturer. From their work, it is not clear what would stop the user from updating blockchain data without receiving the part? The confirmation of receiving the part or not confirming is not giving much to the security. Their solution is solving mostly audit and integrity problems, and the method works best for a genuine check after the manufacturing or during usage of the part.

Many other works [6, 7, 11] describe how the user uploads a file to the cloud and how the design is protected using digital rights management (DRM); however, they do not present a viable solution to protect the copyright of 3D files in the long term. These works do not describe how to protect files from being exposed to 3rd parties nor the possibility to detect and prevent illegal parts manufacturing. Our solution proposes a real solution to enforce copyright protection, illegal parts detection, and prevention of manufacturing.

While it is essential to allow users and manufacturers to determine if any restrictions exist on reproducing a 3D object, ideally there must also be a mechanism in place to prevent the unauthorized reproduction of the 3D object, primarily when the 3D design represents an illegal or dangerous part, like a firearm. As the 3D file itself representing the 3D object according to this scenario does not necessarily have any means attached preventing unauthorized use of the

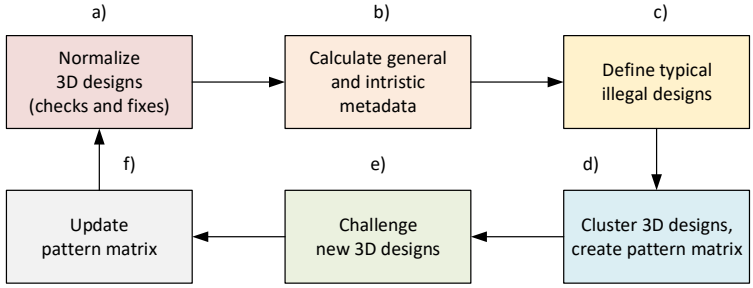


Fig. 1. The proposed method concept

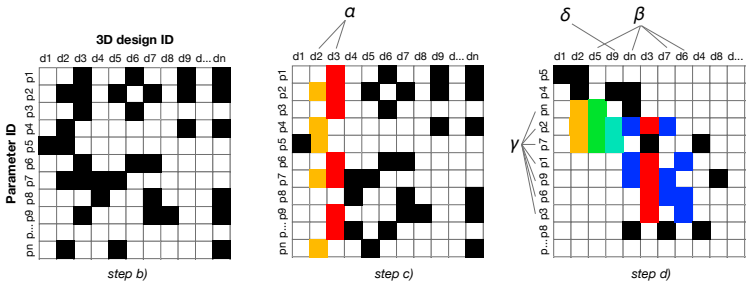


Fig. 2. *step b)* - Calculate general and intrinsic metadata, and store into binary matrix, *step c)* - Define typical illegal objects, *step d)* - Cluster 3D designs and produce pattern matrix, α - illegal 3D designs that were initially marked, β - adjacent illegal designs similar to the ones initially marked, γ - parameters that unify the illegal designs, δ - the design in-between two groups, not strictly differentiated to any of the major groups.

3D file, the known methods cannot be used. The authorization means must be integrated with the manufacturing device itself, e.g., before the start of each manufacturing job, the manufacturing device needs an authorization from the rights holder or confirmation that no restrictions exist. However, with the use of our cloud manufacturing operating system and ecosystem [29], the detection and protection from illegal parts manufacturing can be performed in the cloud. In the case of our proposed method, there is no need for significant modifications on the AM machine side to support a safe and legal method to produce parts.

3 Proposed method

To achieve the detection of 3D designs IP copyright infringement and illegal physical objects in 3D printing, we use pattern recognition based on the seriated

matrix of 3D designs and their important parameters. The proposed concept relies on several important steps depicted in Figure 1:

a) *Normalize 3D designs*, this steps includes 87 different checks and fixes. We check the CAD file consistency on three different levels: file format level, 3D design mathematical consistency level, and physical consistency level, e.g., watertight, wall thickness within a certain threshold, and many more. We mathematically reassemble the model [19, 28] and fix non-manifold edges, remove duplicate faces, remove hidden malicious geometry, and more.
b) *Calculate general and intrinsic metadata* to find important parameters to be used in clustering and pattern recognition. Create initial matrix of objects and parameters, presented on Figure 2, *step b)*. We extract more than 150 different *general parameters* from the CAD design, e.g., scale factor to find the original measurement units, number of triangles, bounding box enclosing the object, maximum outer dimensions, volume, shadow volume, voxelized shape, center of mass, skeleton, corpus indicator, detail indicator, deviation of angles, bounding corners density, average deviation of points, and many more.

   Additionally, we extract parameters intrinsic to IP, which should be protected, and illegal objects to detect, e.g., firearms. For example, to find *intrinsic parameters* for firearms, we perform analysis of adjacent faces of a 3D design, so that faces lie on the open cylinder surface of the same cylinder with a certain threshold. We can extract the number of cylindrical shapes, the diameter of cylindrical shapes, the height of cylindrical shapes, the number of triangles participating in cylindrical shapes, the ratio of missing triangles in a cylindrical shape, and many more. For the pattern matrix and the step *d)* below, we would need a binary representation of parameters. Thus, from 150 parameters, we get more than 3750 by discretizing non-binary parameters into categories.
c) *Define typical illegal designs* as a part of a supervised machine learning process. Later, in the process, these parts will become indicators or contrasting bodies, to detect groups and classes we are looking for, depicted in Figure 2, *step c)*. For our experiments, we mark objects, which we already know are firearms or contain firearm parts or firearm inverted parts, e.g., a section of an injection mold for firearm production.
d) *Cluster 3D designs* into loose groups without strict differentiation, forming a pattern matrix. We are using conformity and seriation [21, 22, 32] to calculate our pattern matrix. Then, we perform the analysis of the matrix in automated mode to find classes with a certain threshold. However, the pattern matrix would provide an interesting insight to analyze the data visually. Our method ensures simultaneous pattern discovery at several information levels - from local patterns to global [21,22]. Initial matrix example is shown on Figure 2, *step b)*, seriated and transformed matrix example is shown on Figure 2, *step d)*.
e) *Challenge new 3D designs* against the pattern matrix. Firstly, a newcomer 3D design is normalized in step *a*), then based on the metadata calculated in

step *b*), we perform a fast classification by matching with other 3D designs, which already positioned in the pattern matrix and have close parameters. Finally, we can understand which class of objects it belongs to. Copyright holders can run different types of searches and investigations for the copyright infringements. For example, it is easy to perform a quick check of whether the object fits into one of the classes of illegal objects. Challenging a new 3D design does not require a recalculation of the pattern matrix.

f) *Update pattern matrix* with the new designs. When there is a considerable amount of new 3D designs within a certain threshold, then the pattern matrix will be recalculated to accommodate new designs. Adjacent designs within a certain threshold could automatically or with operator supervision be marked as common illegal objects. Then, the process repeats over and over again. Over time, the system can adjust for new types of IP to be protected or new types of illegal objects, e.g., new shapes of firearms.

Similarly, the solution could be implemented inside the firmware of the AM machine, like a 3D printer. As an option, an exported pattern matrix can be stored on a hardware chip to help AM machines to challenge 3D designs sent to a job queue. The possibility for a fast classification, whether it is an illegal object to manufacture, would not allow the AM machines to produce illegal or dangerous parts.

4 Evaluation

A software architecture used to perform the evaluation of the proposed method is depicted in Figure 3:

a) *Commander module* is used to launch the workflow described in the proposed method concept Figure 1. The module consists of Python scripts that send commands and files to *Normalizer module* and *Clustering module*.

b) *Normalizer module* is used to normalize 3D designs, calculate general and intrinsic metadata, and challenge new 3D designs against the pattern matrix. It stores 3D designs to Hadoop Distributed File System (HDFS) [26].

c) *Extracted features database* keeps the general and intrinsic metadata on all 3D designs. We use Apache Cassandra [18] as it is a highly available, scalable, and fault-tolerant column-oriented key-value storage with the ability to store over two billion values per row. It is easy to store pattern matrix and perform CQL [9] queries to match new 3D design parameters against pattern matrix. We are also able to keep several versions of pattern matrices in the Cassandra database.

d) *3D design storage* is an HDFS file storage used by the Normalizer module to initially store processed files and then revisit the files if new parameters are added.

e) *Clustering module* is implemented as the Hadoop Map/Reduce Job. It is important to use parallel Map/Reduce implementation of conformity calculation and seriation. Conformity calculation and seriation sequential algorithms implementation require an exponential increase in computational

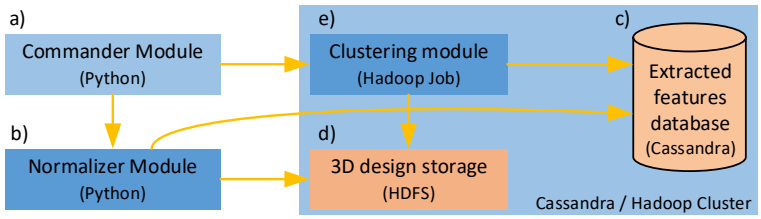


Fig. 3. Architecture of the proposed method

time for binary matrices larger than 10 x 10. Our Hadoop Map-Reduce parallel implementation of conformity calculation significantly outperforms sequential algorithms.

This module is responsible for updating the pattern matrix in case of the number of new designs reaches a threshold preset in the settings file.

Experiments were performed on a test database with 5831 design files in STL [27] format. 3D design files were selected from Thingiverse [8] and other sources [1, 3, 4]. Test database included 3D designs from different areas of life: animals, architecture, kitchen appliances, toys, movie characters, vehicles, interior objects, firearms, and many others.

Normalization stage with checks and fixes was implemented on Python language with the usage of CGAL [10] library. As a next step, all files were run through a batch job to find general and intrinsic metadata parameters. We were able to obtain 3728 parameters for the test dataset.

We performed *normalization* stage on a 4 core, 8GB ram, 32GB SSD Azure VM instance, compute optimize type (Standard_F4s_v2). On average, it took 2.31s for 1MB, 10.46s for 5MB, 48.14s for 20MB, 283.58s for 100MB, 35 min 36s for 500MB, and 1h 48min 12s for 1Gb respectively. Total, all file size was 6.19GB, and the total time for 5831 files was 6h 37min 34s.

For the experiment, we selected 28 different indicative objects containing firearm shaped parts and marked those 28 designs as common illegal objects for our algorithm.

Then we run the algorithm to classify 3D objects based on the indicators and parameters which should help to detect firearms. We have implemented Conformity calculation [20, 32] and matrix seriation [21, 22] using Hadoop Map-Reduce framework [31]. We run Map-Reduce jobs on a cluster of 4 machines in Microsoft Azure cloud [2], each machine having 4 cores, 8GB RAM, 30GB SSD. It took 12 minutes to perform pattern matrix calculation based on the test data.

Each new 3D design file pattern recognition (search by the parameters in the pattern matrix) took 0.32 seconds on 4 core, 8GB ram, 32GB SSD Azure VM instance.

For new incoming illegal 3D designs tested against the pattern matrix, we were able to obtain a very close class of objects similar to initially selected

firearm 3D designs. Example visualization of correctly detected Liberator barrel is shown in Figure 4. 3D design in-between two groups, not strictly differentiated as a firearm, the test visualization is shown in Figure 5. Something which was not detected as a firearm example is shown in Figure 6.

Further tests showed that the solution would find not only a direct shape of the firearm but also an inverse shape, like an injection mold, or a tool to create a firearm.

This approach, similarity, could be extended to any type of multimedia, not just 3D objects.

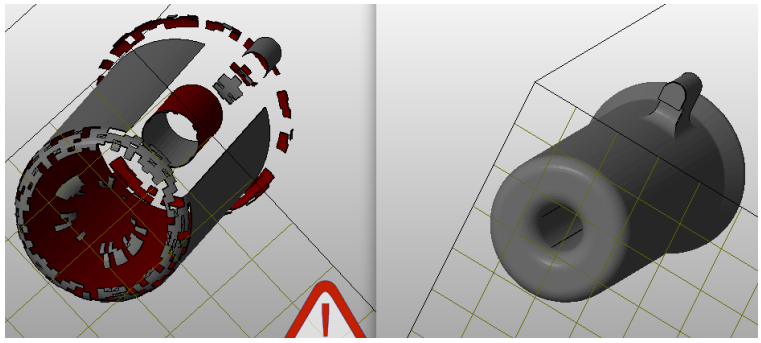


Fig. 4. The Liberator barrel detected as a part of a firearm

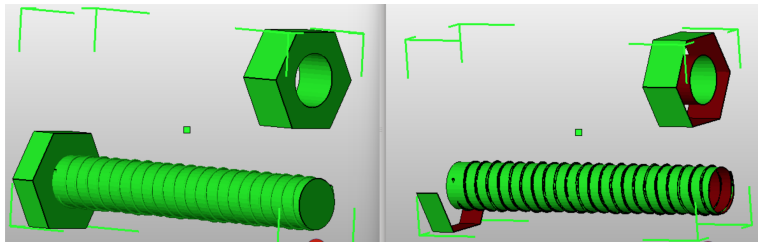


Fig. 5. 3D design in-between two groups, not strictly differentiated as a firearm

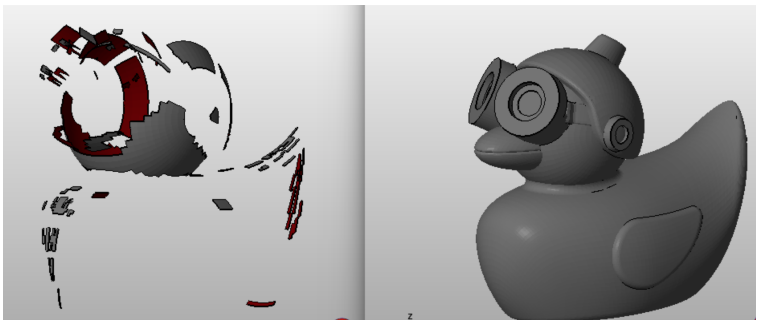


Fig. 6. 3D design that was not detected as a firearm

5 Conclusion

We believe that in order to detect and withstand the global threat of IP copyrights violation efficiently, firearms spreading, illegal 3D objects manufacturing, especially firearms manufactured with the help of AM and 3D printing particularly, we need preventive and detective controls on software, firmware, and hardware levels. The method to protect intellectual property (IP) in automated manufacturing (AM) presented in this paper is based on the radical improvement of preventive and detective controls on software, firmware, and hardware levels. These controls would help to find potential cases of copyrights infringement, illegal objects, and firearms manufacturing with the use of automated manufacturing (AM) machinery. The manufacturing of illegal physical objects on a software level could be prevented through the cloud manufacturing operating system controls, which would not allow sending a manufacturing file to an AM machine. On the machine level, the prevention of manufacturing of illegal parts cloud be implemented through encoding the proposed method into the AM machine firmware, or into a chip integrated into the hardware.

In future research, interesting enough will be to perform experiments on a bigger dataset of 2.14M files and extend the presented approach to a fast 3D design search.

References

1. Defense distributed. https://defcad.com, accessed: 29-09-2019
2. Microsoft azure. http://azure.microsoft.com, accessed: 25-09-2019
3. Thingiverse. https://www.thingiverse.com, accessed: 29-09-2019
4. Youmagine. https://www.youmagine.com, accessed: 28-09-2019
5. All3DP: 2019 3d printed gun digest: All you need to know. all3dp.com (July 2019), https://all3dp.com/1/3d-printed-gun-firearm-weapon-parts/

6. Astovasadourian, A., Naro, O., Cabanel, V.: 3-d printing protected by digital rights management (Apr 20 2017), uS Patent App. 14/950,431
7. Badhani, H., Chopra, A., Goel, N.P., Panda, A.S.: Method and apparatus for controlling printability of a 3-dimensional model (Oct 4 2016), uS Patent 9,457,518
8. Buehler, E., Branham, S., Ali, A., Chang, J.J., Hofmann, M.K., Hurst, A., Kane, S.K.: Sharing is caring: Assistive technology designs on thingiverse. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 525–534. ACM (2015)
9. Chebotko, A., Kashlev, A., Lu, S.: A big data modeling methodology for apache cassandra. In: 2015 IEEE International Congress on Big Data. pp. 238–245. IEEE (2015)
10. Fabri, A., Teillaud, M.: Cgal-the computational geometry algorithms library. In: 10e colloque national en calcul des structures. p. 6 (2011)
11. Glasgow, D., MacLaurin, M.B., Sherman, C.E., Ramadge, D.: Digital rights and integrity management in three-dimensional (3d) printing (Mar 14 2017), uS Patent 9,595,037
12. Holland, M., Nigischer, C., Stjepandić, J., Chen, C.: Copyright protection in additive manufacturing with blockchain approach. Transdisciplinary Engineering: A Paradigm Shift **5**, 914–921 (2017)
13. Hou, J.U., Kim, D., Ahn, W.H., Lee, H.K.: Copyright protections of digital content in the age of 3d printer: Emerging issues and survey. IEEE Access **6**, 44082–44093 (2018)
14. Isbjornssund, K., Vedeshin, A.: Method and system for enforcing 3d restricted rights in a rapid manufacturing and prototyping environment (Feb 27 2014), uS Patent App. 13/973,816
15. Isbjörnssund, K., Vedeshin, A.: Secure streaming method in a numerically controlled manufacturing system, and a secure numerically controlled manufacturing system (Dec 3 2015), uS Patent App. 14/761,588
16. Kennedy, Z.C., Stephenson, D.E., Christ, J.F., Pope, T.R., Arey, B.W., Barrett, C.A., Warner, M.G.: Enhanced anti-counterfeiting measures for additive manufacturing: coupling lanthanide nanomaterial chemical signatures with blockchain technology. Journal of Materials Chemistry C **5**(37), 9570–9578 (2017)
17. Kietzmann, J., Pitt, L., Berthon, P.: Disruptions, decisions, and destinations: Enter the age of 3-d printing and additive manufacturing. Business Horizons **58**(2), 209–215 (2015)
18. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Systems Review **44**(2), 35–40 (2010)
19. Leong, K., Chua, C., Ng, Y.: A study of stereolithography file errors and repair. part 1. generic solution. The International Journal of Advanced Manufacturing Technology **12**(6), 407–414 (1996)
20. Liiv, I.: Visualization and data mining method for inventory classification. In: 2007 IEEE International Conference on Service Operations and Logistics, and Informatics. pp. 1–6. IEEE (2007)
21. Liiv, I.: Pattern discovery using seriation and matrix reordering: A unified view, extensions and an application to inventory management. TUT Press Tallinn (2008)
22. Liiv, I.: Seriation and matrix reordering methods: An historical overview. Statistical Analysis and Data Mining: The ASA Data Science Journal **3**(2), 70–91 (2010)
23. Lin, N.H., Huang, T.H., Chen, B.Y.: 3d model streaming based on jpeg 2000. IEEE Transactions on Consumer Electronics **53**(1) (2007)
24. Mattingly, T.D., Tovey, D.G., O'brien, J.J.: System and methods for three dimensional printing with blockchain controls (Sep 13 2018), uS Patent App. 15/913,382

25. Sepp, P.M., Vedeshin, A., Dutt, P.: Intellectual property protection of 3d printing using secured streaming. In: The Future of Law and eTechnologies, pp. 81–109. Springer (2016)
26. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The hadoop distributed file system. In: 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). pp. 1–10. Ieee (2010)
27. Stroud, I., Xirouchakis, P.: Stl and extensions. Advances in Engineering Software **31**(2), 83–95 (2000)
28. Szilvśi-Nagy, M., Matyasi, G.: Analysis of stl files. Mathematical and Computer Modelling **38**(7-9), 945–960 (2003)
29. Vedeshin, A., Dogru, J.M.U., Liiv, I., Draheim, D., Ben Yahia, S.: A digital ecosystem for personal manufacturing: An architecture for cloud-based distributed manufacturing operating systems. In: Proceedings of the 11th International Conference on Management of Digital EcoSystems. p. 224–228. MEDES '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3297662.3365792, https://doi.org/10.1145/3297662.3365792
30. Vedeshin, A., Dogru, J.M.U., Liiv, I., Yahia, S.B., Draheim, D.: A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. IEEE Access (2019)
31. Verma, J.P., Patel, B., Patel, A.: Big data analysis: recommendation system with hadoop framework. In: 2015 IEEE International Conference on Computational Intelligence & Communication Technology. pp. 92–97. IEEE (2015)
32. Võhandu, L.: Fast methods in exploratory data analysis. Transactions of TTU **705**, 3–13 (1989)
33. Walther, G.: Printing insecurity? the security implications of 3d-printing of weapons. Science and engineering ethics **21**(6), 1435–1445 (2015)

Appendix 4

IV
P.-M. Sepp, A. Vedeshin, and P. Dutt. Intellectual property protection of 3d printing using secured streaming. In *The Future of Law and eTechnologies*, pages 81–109. Springer, 2016

Intellectual Property Protection of 3D Printing Using Secured Streaming

Paula-Mai Sepp, Anton Vedeshin, and Pawan Dutt

Abstract 3D printing technology is a new and emerging technology which is capable of changing the world. However, an easy access to 3D printing technology makes a convenient way to illegally reproduce physical objects regardless of copyrights, license, and royalty payments. As 3D printing of physical things at home might become the "new normal," it will pose threats to traditional intellectual property laws, which were created in an era when copyright infringement of physical objects, or also defined as "physibles," was yet to come. The authors have brought forward the legal issues and have attempted to describe a unique technical solution—secured streaming which solves or at least partially solves the problem of copyrights in 3D printing. The proposed solution provides a possibility for a copyright owner to limit the number of 3D prints. He can specify the number of copies that are allowed for the manufacturer or an end user to produce. Moreover, secured streaming has detective and protective controls to detect information system compromises and to stop streaming of 3D designs to 3D printers.

1 Introduction

Three-dimensional (3D) space printing technology is often referred to as the new hot and emerging technology, capable of changing the world. In fact, the roots of the technology reach back to the late 1970s, when the seed for additive manufacturing

P.-M. Sepp (✉)
Ministry of Justice of Republic of Estonia, Tõnismägi 5a, 15191 Tallinn, Estonia
e-mail: paula.sepp@outlook.com

A. Vedeshin
3DPrinter OS, Mektory Innovation Center Building, Raja 15, 12618 Tallinn, Estonia
e-mail: anton.vedeshin@gmail.com; http://www.3dprinteros.com

P. Dutt
Tallinn Law School, Tallinn University of Technology, Akadeemia tee 3, 12618 Tallinn, Estonia
e-mail: pawan.dutt@ttu.ee

T. Kerikmäe, A. Rull (eds.), *The Future of Law and eTechnologies*,
DOI 10.1007/978-3-319-26896-5_5

idea was first put down as a joke, in a newspaper article by David Jones.[1] An independently filed patent application by Wyn Kelly Swainson for the same technology was granted in 1977.[2] 3D printing can be described as a method of joining materials, layer by layer, on the basis of a computer automated design (CAD) model or 3D-scanned file.[3] If inkjet printers print pixels from the screen onto a piece of paper using ink on an XY-axis, then 3D printers print using plastic string on an XYZ-axis, making the object three dimensional.[4] 3D printing technology is on the verge of a breakthrough into home use and is revolutionary in the sense that it enables everyone to become creator of things.[5] Additive manufacturing enables designers to create products with complex shape and very small detailing, which have previously been hard to execute with other methods of manufacturing.

3D printing is thus one of the automated manufacturing methods to produce physical objects by adding material layer by layer. There are many good examples of using 3D printing technology in industries and small and medium enterprises. New Balance is printing shoes by the size and exact shape of a sportsman's feet.[6] Francis Bitonti, a famous New York designer, is printing exceptional dresses and home accessories.[7] At remote locations, like aircraft carriers, oil derricks in sea, and space stations, it is important to get printable parts at the point of need and time of need without extra costs for logistics and shortest lead time. Boeing and Airbus are printing turbine parts to increase efficiency and reliability of aircraft engines. However, 3D printing is not anymore a method for prototyping at big factories and corporations. 3D printers are not yet at everyone's home, but even today they are at least accessible within walking distance in any major city. An easy access to 3D printing technology makes a convenient way to illegally reproduce physical objects regardless of copyrights, license, and royalty payments. After obtaining a printable 3D design, it can be reproduced many times without the possibility for a copyright owner to trace.

New digital technologies have made copying a lot easier than it has been before, and we have already witnessed the collateral damage in relation to copying of music and movies. As 3D printing of physical things at home might become the "new normal," it will pose threats to traditional intellectual property laws, which were created in an era when copyright infringement of physical objects, or also defined as

[1] Bradshaw et al. (2010), pp. 7–8.

[2] Ibid.

[3] Stahl (2013), pp. 3–4.

[4] Howells (2014), p. 13.

[5] Weinberg (2013), p. 1.

[6] New Balance (2013). Press release: New Balance Pushes the Limits of Innovation with 3D Printing. Available at: http://www.newbalance.com/press-releases/id/press_2013_New_Balance_Pushes_Limits_of_Innovation_with_3D_Printing.html (accessed 20.08.2015).

[7] See examples of high-end 3D designs from Francis Bitonti Studio web page. Available at: http://www.francisbitonti.com/ (accessed 20.08.2015).

"physibles,"[8] was yet to come. What also makes 3D printing stand out is the speed to market—it enables people to scan and create a physical product in a matter of hours. In particular, the intellectual property issues are paramount in relation to copyrights because they are free and exist automatically for a work that has been fixed in a tangible form. Other forms of intellectual property are not left untouched, as problems will also arise in the field of patents, trademarks, and industrial design protection. Another reason why copyrights have the biggest likelihood of becoming the object of infringement is that most items available for home 3D printing include designs of decorative nature or fan fiction art, which does not entail a useful feature.[9] The leading approach to 3D printing originates from the U.S., because the legal side of 3D printing has been dealt with more extensively there. The main elements of 3D printing technology are the physical 3D object and digital CAD files, which can be obtained through designing process in a CAD software or by 3D scanning. The digital CAD file and the physical 3D printed object easily meet the fixation requirement of copyright protection.[10] It is fundamental to recognize that CAD files differ from MP3 files used as music carriers and MP4 files used for audiovisual content, for which the suitability for intellectual property protection is not under doubt.[11] Because 3D printing contains both digital and physical characteristics, it is hard to determine whether the main characteristics and related intellectual property issues should be evaluated separately or as a whole. Scholarly opinions are roughly divided into two in deciding whether the CAD file or the 3D printed object poses a more pivotal question for the suitability for intellectual property protection.[12]

Gartner estimates that by 2018, 3D printing will result in the loss of at least $100 billion per year in intellectual property globally.[13] This creates a completely new problem of copyright protection, as it is relatively easy to copy and reproduce objects, and in some cases 3D printing even creates a threat on the society if parts are produced from not original or compromised designs.

[8] The online peer-to-peer sharing site, The Pirate Bay, launched a category for 3D designs called "physibles." See, for example: Walters (2012).

[9] Doherty (2012), p. 358.

[10] Dasari (2013), p. 279.

[11] Twomey (2014), p. 33.

[12] See: Dolinsky (2014), pp. 629–631. According to Dolinsky, there is no question in the copyrights of 3D printed objects, which are protected as "pictorial, graphical and sculptural works," and the main question will be the copyrightability of CAD files. See also: Rideout (2011), pp. 167–168. Rideout on the contrary states in his work that the copyrightability question of a CAD file is conditional to the eligibility of copyright protection of the 3D printed object. According to Rideout, it is the CAD files that would likely fall under "pictorial, graphic and sculptural works" and more specifically under "technical drawings, diagrams and models."

[13] Gartner (2013). Press release: Gartner Reveals Top Predictions for IT Organizations and Users for 2014 and Beyond. Available at: http://www.gartner.com/newsroom/id/2603215 (accessed 20.08.2015).

In this chapter, the authors will look into the legal issues and will attempt to describe a technical solution—secured streaming which solves or at least partially solves the problem of copyrights in 3D printing. The solution provides a possibility for a copyright owner (CO) to limit the number of 3D prints. CO can specify the number of copies that is allowed for the manufacturer or an end user to produce. Moreover, secured streaming has detective and protective controls to detect information system compromises and stop streaming of 3D designs to 3D printers.

2 Why Protecting Printable 3D Designs Has Become So Important

One would ask why copyright protection in automated manufacturing and 3D printing particularly is so important. Society has lived a long time without a special solution or just using Digital Rights Management (DRM) to secure CAD designs.

About 60–70 years ago we were at "paper age"; most of the products' technical drawings were done on paper. Imagine an individual who wants to copy the product; he makes pictures of the sketches. Now he needs to find a production technology, train engineers, set up a factory and production lines to produce prototypes and then a real product. Let's assume this would take around 2 years.

Then about 15–20 years ago we entered the digital age, which offered us the use of CAD tools. However, these tools were used mostly to create a virtualization of a product to make right measurements, different types of simulations, quicker changes to the structure after prototype testing cycles. If somebody would get such CAD design, he would still need to find a production technology, train engineers and set up a production facility; compared to the paper age example, this would take half a year to produce a real product.

Nowadays, at 3D printing age, CAD intended for 3D printing already has all important information inside to produce the real object. If one would compromise such a design, he can get to the market with the product in just a few days. As usually, the 3D design intended for 3D printing has all the needed information to manufacture a product according to all specifications, tolerances, durability and taking into account force distribution and dispensation.

3 What Are Current Technical Solutions Stating?

With the development of technology, the security requirements evolve too. Most of 3D printers connected to networks do not have enough protection against today's threat of digital theft. See Fig. 1 for a detailed 7 action (A1 . . . A7) and 6 transition (T1 . . . T6) steps of a CAD design life cycle from a product idea to a physical object

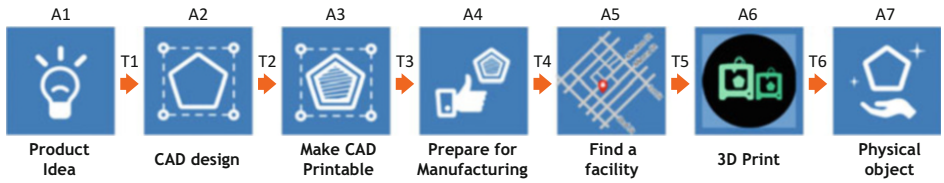


Fig. 1 7 action and 6 transition steps of a CAD design lifecycle: from a product idea to a physical object

manufacturing process. This is an illustrative model to represent the life cycle of a CAD design and in practice could contain more or less steps from product idea to a physical object. At every action or transition step, there is a special CAD or other software involved, and at transition steps in-between action steps there are different types of storage or information transfer technology used.

Action steps A1, A2, A3, A7 are possible to secure using different types of cloud CAD solutions from numerous companies[14]; however, none of these software packages offer end-to-end functionality or integrated security for the whole life cycle of a printable 3D design, which makes them vulnerable at least at steps A4, A5, A6, A7 and T3, T4, T5. Users use email, USB sticks, SD cards, network drives to share CAD designs with their colleagues or partners at steps T3, T4, T5 and frequently at steps T1, T2.

Transition steps T1, T2, T3, T4, T5 are possible to secure by existing technologies; however, most of the solutions are based on DRM model, which is vulnerable by its nature, or even if it is secured during the transition, it is still vulnerable at the end of transition step' for example, in sending files using SSL—when the file is received—it could be copied, or for example if file is downloaded from cloud storage, such as Google Drive, Box.com, or Dropbox, in the end of transition—saving to the hard drive of the personal computer—it becomes vulnerable. Some companies implement nondisclosure agreements (NDA) with employees and partners, which will not help either to protect digital content in a long-term perspective.

T5 is usually a USB or local network connection, which in case of modern 3D printers in not ultimately secured.

Some steps still are not possible to ultimately secure or alternatively could be solved at other than software level. T6 is delivery and a handover of a 3D printed physical object to the user. A7 is usage of the printed object by an end user. At T6 and A7 steps, object could be disassembled and scanned or reverse engineered in a different way for further reproducing.

In the following sections of this chapter, we will introduce an innovative method of securely streaming CAD designs seamlessly through A1–A6 and T1–T5 steps of CAD design lifestyle without exposing it to all involved parties.

NDA, DRM, and existing digital media streaming will not help.

[14] For example: Solid Edge from Siemens, Inventor from 3D Systems, Autodesk, Solid Works, etc.

Signing an NDA with employees or collaboration partners is a usual practice used within big, small, and medium-size organizations. Once the digital content is exposed due to intentional or nonintentional disclosure of information, there is no way to stop copying it or getting it back. Thus NDA is not anyhow helping to protect 3D designs long term.

Another method which is frequently used to secure digital content is DRM. Classical DRM works in a way that digital content is encrypted (usually the whole file) and then using email, web browser, CD/DVD disk, USB flash drive, or SD card is sent/given to an end user, who using a key decrypts the digital content and consumes it (please see figure below). If the content is stolen and the key is compromised or calculated (which is just a matter of time and computing power) then the content secured with DRM technology could be copied and used as many times as individuals want, there is no way to stop reproduction of such content. Good examples are numerous software packages, operating systems like Windows, DVD movies, games, MP3 media, etc. This is exactly the content Torrent networks and Pirate Bay like sites are full of. Not only DRM technology is an easy target of intellectual property copyright violation; moreover, malicious software and viruses are distributed together or inside packages with cracked DRM software, movies, and music.

Up to now many companies in digital media sector, such as Netflix, YouTube, Spotify, TuneIn, have used media and secured media streaming. From the first sight it seems that this would probably work for 3D designs too, as 3D designs are media to some extent. Media streaming also protects the content, even better than DRM. At a closer look, in order to stream media, it should be in streamable format, and information should not be anyhow available to download or obtain in a different way prior to digital media consuming session. Copyright protection is achieved by the complication of getting the stream, converting it to a different file format, and then distributing it. Sometimes the quality of media is not there, so it does not make any sense to grab, for example, a video from YouTube, or it is too complicated for the end user to go through all procedures to store one song or a movie to the hard drive of his computer; in 95 % of the cases it is easier to pay, as in case of streaming media businesses' business models, the subscription price is comparably low, compared to the cost of ownership of a song album or a DVD with movie. Many businesses use nowadays media streaming (not necessarily secured) as a sort of protection; most of the people pay for the service, and of course there is a minority who still can compromise the stream and copy the content. Another problem is that the content should be possible to find. Assume there is an individual who decoded and copied the song or a movie from stream of YouTube, where should he put it so other people can easily find it? Potential consumers of this media probably already have subscriptions to the services if they consume a lot; if not, then even 5–10 % of the cases will not affect the revenue and loss of royalty payments much. Many software companies went similar way; for example, Microsoft Office 365 allows you to use this product on a monthly basis, without need to own a license and with the possibility to opt out any time.

Why could the same or similar media streaming technologies not be applied to 3D printing? There is a huge difference in the requirements for streaming between printable 3D files and media like movies or songs. In case of streaming video or music, if the stream got recorded this is not a big problem; as we described above, such copy will have quite a short life in torrent networks, and the same user will watch or listen to it for hundreds/thousands of times; in case of 3D printable object, this could be massively reproduced by a first-hand consumer. Another important difference is connected to the quality of stream; for example, if during the movie-watching process few frames would drop or music will skip half a second few times, this is not a big deal. However, in case of the stream going to 3D printers, every byte and every bit should be delivered and in the right sequence. In the best case, user will get bad quality product and probably reprint it or use a competitive product. The worst case, as 3D printers are very precise and expensive machines, wrong sequence of codes or some bytes missing can break the 3D printer. In comparison with video and music—it is not possible to break a TV or a monitor with the wrong video/music stream. Finally even the worst case, 3D design could be compromised on the storage server or on the way to 3D printer; if it is a mission critical part, for example airplane turbine part, then the authors believe the reader of this book by all means would not like to be a passenger of that airplane. In comparison with video and music—the end user will soon understand that actors in the movie are different or that the song is performed by, for example, Jennifer Lopez. These points have the biggest impact on the fact that video or music streaming is not a suitable solution, and there is a space for next generation streaming solution, which delivers right data, at the right place, in the right time without compromising security or, even better, with ultimate security.

4 Understanding the Legal Aspects of 3D Printing

As 3D printing is a completely unregulated field, regulating it will pose different challenges to legislators because the implementation of any regulative measure can have unforeseeable effects to further developments of 3D printing technology. The possible options of regulation may entail in the enforcement of hard regulation by the state, community self-regulation, or leaving the industry unregulated for as long as possible and pose regulations only after the industry has had time to mature. The existing legislation for copyrights usually involves an unlimited list of protected works, which allows for the interpretation of new technologies and mediums under copyright protection, meaning that drafting a specific regulation for 3D printing is not essential in such an early phase of technological developments. Nevertheless, scholars have proposed different existing categories of work, which could be treated as analogs to 3D printing, for the reason that it has not been explicitly regulated. Examples on how to deal with the rapid technological developments of 3D printing from the legal perspective can be found in case law dealing with disruptive technologies that have changed the world.

Experience in the United States and the United Kingdom (where many of these issues have been dealt with first) has shown that attempts to ban hardware or to outlaw devices are especially problematic. This is certainly true for those devices which have legitimate as well as illegitimate applications. This is very often the case where the hardware is distributed through distributors who are not in a position and have no effective means of monitoring usage by an end user of the devices.[15] Often, copyright exceptions are carved out for private users as a way of conciliation of interests of copyright owners, the equipment industry, and ultimately the consumers. This is to ensure that creators are rewarded but not at the cost of disadvantaging consumers in an unreasonable manner.[16]

Historically speaking, it could be said with some conviction that the law of copyright owes its development to significant advances in technology, starting from the invention of the printing press itself. However, it should also be noted that the judiciary has preferred to defer to the wishes of the legislature in this regard and has been consequently hesitant to expand protections under the copyright regime without explicit guidance from Parliament. This view has gained traction all the more for reasons that the legislature alone has constitutional authority and institutional ability to take into account the various competing interests in society which tend to surface every time a new path-breaking invention is brought into commercial existence.[17] It is important for the law to encourage innovation and to invigorate commercial activities, rather than sacrificing the above ideals on the ground of mere possibility of misuse of new technology to the detriment of some copyright owners.[18] After all, it must also be understood that the Charter of Fundamental Rights recognizes the freedom to conduct business.[19]

Fair dealing provisions are useful in this regard as they generally provide important limitations to owner's rights (for the purposes of noncommercial research or private study, critical reviews, and news reporting).[20] However, in most jurisdictions these provisions are fairly restrictive, unlike under the United States law.[21] It is important to note that in the United States, only guidelines regarding fair use are provided, and these apply to all types of work, although this can be controversial.[22] The role of transformative use (i.e., making a new work by adding newness,

[15] Copinger and Skone James (2005), p. 1452.

[16] Xiaoxiang Shi (2012), p. 533.

[17] Sony Corporation of America v Universal City Studios, Inc., 464 U.S. 417 (1984), p. 431.

[18] Merges et al. (2012), p. 720.

[19] Nyman-Metcalf et al. (2014), p. 37.

[20] Copinger and Skone James (2005), p. 481.

[21] Copyright Act 1976, 17 U.S.C., Section 107.

[22] Copinger and Skone James (2005), p. 481, and also see fn. 14 on that page where criticisms regarding the US approach and their contrast with the principle of statutory construction noscitur a sociis (i.e., that the meaning of a doubtful word may be ascertained by referring to the meaning of words associated with it) is discussed.

either in purpose or character) is also important for furthering the cause of copyright law through the implementation of the fair use doctrine.[23]

It should always be borne in mind that the concept of vicarious liability has given rise to complications under copyright laws since this branch of law rarely renders anyone expressly liable for infringement activities committed by another. The doctrine of "contributory infringement" is after all "....merely a species of the broader problem of identifying the circumstances in which it is just to hold one individual accountable for the actions of another."[24] Sale of articles which can be used for infringing as well as other and lawful uses is not sufficient to render the seller as a contributory infringer since such an absurdity would "block the wheels of commerce."[25]

Also of interest is the "Staple Article of Commerce" doctrine, which emphasizes upon the rights of others to engage in commerce which is of such a nature as being substantially unrelated with infringement of an owner's copyright. Thus, a product which can be used widely for legitimate and unobjectionable purposes and which is capable of substantial noninfringing use would not come under the purview of the doctrine of contributory infringement.[26] Time and again various authors have reasoned that copyright should not be stated as violated if new technologies are developed which possess both types of applications—namely, infringing and noninfringing.[27] This is especially the case when the courts must assess the public interest in accessing that article of commerce while deciding on the merits of the matter.[28] This should, however, not be confused with the inducement rule, whereby one who distributes a device with the sole objective of promoting infringement of copyright becomes in turn liable for the infringing acts of third parties.[29] Although the "Staple Article of Commerce" has its roots in patent law (whereby distribution of a component of a patented device will not lead to infringement of the patent, provided that it is suitable for other uses), it is noteworthy that the aforementioned defense (used successfully in the Sony case outlined below) is not absolute, and indeed the courts now also consider whether the infringing activity outweighs the noninfringing activity.[30]

[23] Khaosaeng (2014), p. 241.

[24] Sony Corporation of America v Universal City Studios, Inc., 464 U.S. 417 (1984), p. 435.

[25] Ibid, p. 441.

[26] Ibid, p. 442.

[27] For example, see Raval (2012), p. 98, where controversies regarding gaming consoles and rights of gamers to make modifications in the software are explored in the prism of dichotomies under US and Australian copyright laws.

[28] Merges et al. (2012), p. 363.

[29] As held in Metro-Golwyn-Mayer Studios Inc. v. Grokster, Ltd. Supreme Court of the United States 545 U.S. 913 (2005).

[30] Haque (2008), p. 377. where the author discusses the case Metro-Goldwyn-Mayer v Grokster (9th Cir) 380 F.3d 1154 (2004).

4.1 Case Laws Dealing with Disruptive Technologies

It would be of interest to see how the courts have dealt with issues regarding technological progress in the face of copyright law concerns in the 1980s and 1990s, in order to seek to foretell where the issue of 3D printers is headed. Attempts to outlaw video recorders, tape-to-tape recorders, and MP3 players will be studied, since each represents an advancement of technologies lying at the intersection of computer technology and the Internet frontier. Efforts to restrict the above devices (and their leapfrogging technology enablements) have consistently failed to fructify, and it should be of no surprise that 3D printing devices too will face similar birth pangs (in issues concerning legality thereof).

4.1.1 The Sony Case: Time Shifting[31]

This United State's Supreme Court case from the early 1980s dealt with home video tape recorders. The legal issue which was raised here was regarding the sale of copying equipment (namely, Betamax video tape recorders) by the petitioners to the general public and the perceived sense of consequent violation of copyright vested in the respondents. The respondents commenced the proceeding in the District Court by contending that some individuals had infringed the respondent's copyrights by using the Betamax tape recorders to record copyrighted works which had been exhibited on commercially sponsored television. Interestingly, the respondents sought no relief against the Betamax consumers per se. Rather, in an unprecedented move, they sought to impose liability upon the distributors of copying equipment by making the petitioners liable for the copyright infringement by their customers on the ground that the marketing style and process of the Betamax machines by the petitioners was at fault. The respondents thereby sought monetary damages and an equitable accounting of profits, coupled with injunctions against the manufacture and marketing of the Betamax machines.

Underlying the tensions in this particular case was the novel concept of "time shifting" which had been propagated by the petitioners. Time shifting was designed to help average members of the public to use Betamax tape recorders as a means for recording a televised program which he is unable to view at the time of telecast, with the intention to watch the recorded program at a later time. Further, tapes could be reused, recorded programs could be erased, a "timer" function enabled recording of programs from TV when the owner was not at home, and the machines were equipped with a pause button and fast-forward control mechanisms. All in all, this provided a significant leap in the arena of home entertainment systems.

Since the nature of the copying through tape recorders was uniquely private, enforcement of copyright was seen as overreaching and excessive as it would require the monitoring of private behavior and acting against end users who

[31] Sony Corporation of America v Universal City Studios, Inc., 464 U.S. 417 (1984).

committed the above acts in the privacy of their homes. This was also seen as giving rise to a conflict between fundamental human rights and copyright enforcement.[32]

It must be noted that in the 1980s, domestic videocassette recorders were used widely for the purpose of recording broadcasts, despite the fact that it was unlawful to do so. In fact, the laws were amended (no doubt encouraged by the above judgment) to facilitate such recording for purposes of time shifting![33]

What is interesting to consider in this case was the assertion of the United State's Supreme Court that:

There should be a balance between the interests of authors and inventors on one hand and the interest of society in the free flow of ideas, knowledge and commerce. Also of note is the fact that copyright protection never gives the owner complete control over all the possible ways and means in which his work can be used.

The respondents were neither able to prove that the practice of time shifting had caused any impairment to the commercial value of their copyrights, nor could they elucidate (through a preponderance of evidence) upon the potential for harmful effects of this practice in the future.

Nothing should be done to enlarge the scope of the respondents' statutory monopolies under the Copyright Act by means of enjoining the distribution of Betamax tape recorders, collecting sales royalties on above-listed equipment, or other such coercive reliefs. This was especially important since the Betamax tape recorders were held by the court to be "articles of commerce" and consequently were not seen as being subject to copyright law and such attempts by the respondents were seen as an expansion of copyright privileges beyond the limits of the grants authorized by the Legislature.

The noncommercial nature of the use, coupled with the private nature of the recording and playing activity committed entirely within the environs of one's house, readily applied itself to the doctrine of "fair use" of copyrighted works. This sort of activity was seen to be in tune with the legislative goal of serving public interest through open access to information via public airwaves.

The petitioners were merely in the business of supplying a piece of equipment that was generally capable of being used for making authorized or unauthorized copies of copyrighted works and are thus absolved of vicarious liability. What is lacking in this instance is that the petitioner ever had constructive knowledge of the fact that its customers may make use of the tape recording machines for producing unauthorized copies of copyrighted materials. This distinction between copyright and patent laws needs to be stressed upon.

Assuming without accepting that home-use recording of copyrighted material was a form of infringement of copyright therein, an injunction against the Betamax tape recorder would appear to be harsh and inordinate and would result in depriving the public of access to and ability to legally use the machine for the purposes of

[32] Karapapa (2011), p. 257.

[33] Copinger and Skone James (2005), p. 568.

recording noncopyrighted material or material which is capable of being legally copied due to express permission of the copyright owners.

The respondents do not represent all copyright holders, and the petitioners have shown that televised sports events, religious broadcasts, and educational programs comprise a substantial category of copyrighted works—works whose owners welcome the use of Betamax tape recorders for the purposes of legitimate copying of their freely accessible works.

As is the norm, the court took note of surveys, opinion evidence, etc. tendered by the parties to the dispute. These supported the petitioners' claims that substantial numbers of copyright owners did not find the practice of "time shifting" to be objectionable and that harm from "time shifting" is not only highly speculative but also minimal in nature.

Thus, it was held by the Court that sale by the petitioners of such equipment to the public did not constitute contributory infringement of the copyright vested in the respondents.

In hindsight, it can be seen that Hollywood was incorrect when it predicted disaster due to Sony's video tape recorders. Instead, what was noticed is that the movie industry discovered new business opportunities in video rentals and sales. This shows that content industries predict doomsday scenarios on a regular basis when they are confronted with new technologies that threaten existing business models, but subsequently the more resilient businesses find new ways and means to profit from the advancement in technology.[34]

This case (among other notable ones) is the reason why the United States' leadership in the development of new technologies related with time shifting has been globally recognized, and the legal approach adopted by the United States with regard to IP development and consumer rights is often seen to inspire intellectual property (IP) laws enacted in foreign countries (and subsequently eyebrows are raised when the United States' approach is ignored).[35]

However, it must be noted that even in the United States it is now widely acknowledged that the above Sony case did not address the new protections afforded by the Digital Millennium Copyright Act, 1988, and thus equipment manufacturers need to ensure avoidance of a circumvention claim rather than to negate a claim of copyright.[36] Further, the approaches towards this issue have been diluted post 2005, since the Grokster decision. However, a review of the post-2005 period analysis in different countries has proven to be uneven, perhaps being a sign of the far-reaching impact of the Sony decision and the inability of subsequent judgments to completely erase Sony's lucent primacy with respect to developing technologies.[37] Another good indicator is also the Napster case, where the court

[34] Merges et al. (2012), p. 608.

[35] Giblin (2012), p. 639, where the author analyzes the situation in Australia.

[36] Merges et al. (2012), p. 692.

[37] Daly (2007), pp. 319–324, where the author has conducted a review of post-2005 peer-to-peer file sharing issues.

held that the "shifting" analyses of the Sony and Diamond (discussed below) cases were not applicable since in these two cases the methods of shifting resulted in exposure of the material only to the original user and not to the general public.[38]

4.1.2 The Amstrad Case[39]

In the mid 1980s in the United Kingdom, another interesting development in home entertainment music systems took place, pushing the boundaries of "home taping" a step further. Amstrad commenced the manufacture, marketing, and sale of double cassette deck audio systems. The speciality of these systems was that they facilitated the recording from one tape deck to the other at twice the speed of a normal playback, thereby enabling the owner of the machine to copy favorite cassettes at twice the normal playing time. This raised the ire of the majority of record and cassette manufacturing companies, which contended that Amstrad was encouraging home taping of prerecorded cassettes, something which was obviously hurtful to their interests. The owners of the relevant copyrights also sued Amstrad for infringement of copyright in this regard.

Since copyright can be infringed either directly by the infringer or by someone who authorises the infringement, it necessarily thereby follows that a person liable for authorizing infringement will be liable as a joint tortfeasor and also vicariously liable for the acts of his subordinates or agents.[40] Although proof of an act of direct infringement would be required, judicial decisions in this regard are unclear.[41] It was alleged that Amstrad and others were supplying the above equipment in breach of a common law duty of care owed to copyright owners. Further, it was alleged that there was also a breach of an equitable duty of care not to allow goods likely to be used for the purposes of infringement to pass out of Amstrad's hands, without first taking certain necessary and reasonable precautions to ensure that copyrights were not infringed by the usage of such equipment.

However, it should be noted that merely putting into another person's hands the means to do something (which could be infringing or legitimate) is not enough. It should be shown that the supplier has some control over how the means will be used.[42] This is essentially what a grant entails—that the grantor can somehow exercise control over the acts of the grantee.[43] Thus, mere facilitation or giving the users technical means to infringe would not suffice, since users are responsible

[38] Akester (2005), p. 106.

[39] CBS Songs Ltd v Amstrad Consumer Electronics Plc [1988] A.C. 1013.

[40] Copinger and Skone James (2005), p. 449.

[41] Monotti (2013), pp. 325–326.

[42] Yan (2012), p. 123.

[43] Copinger and Skone James (2005), pp. 450–451.

for their own acts (albeit with a few caveats—as such an approach would not work today in a Pirate Bay website type of situation).[44]

One of the interesting causes of action raised in this matter was the offense of incitement to commit offenses under the relevant copyright act, propped up in part on the grounds that the advertising by Amstrad was particularly effective in this aspect and was viewed as encouraging/inciting the general public to buy these machines with the view to copy the contents of their favorite cassettes, thereby breaking the copyright law. The court, however, held that Amstrad could persuade a purchaser to buy a machine through its advertisements but could not possibly influence his decision to infringe copyright.[45]

This case could be seen as a continuation of the rather long and convoluted history wherein the recording industry has tried, without much success, to stop the so-called illicit copying of recordings (as manifested in the present case by tape-to-tape copies). It is interesting to note that the recording industry has targeted not only pirates but also domestic copyists who copy for themselves or people they know.

Interestingly, the House of Lords noted that home copying was widespread, was unpreventable, and brought the law into disrepute, and thus the law should be amended or repealed.[46]

Some of the interesting points noted by the House of Lords in this case were as follows:

The issue of Civil Liability in the form of tort—Even if Amstrad marketed and advertised these equipments in a way which encouraged purchasers to copy their favourite cassettes, thereby giving rise to the accusation of incitement to breach other people's copyrights, none of the parties to the suit were able to prove that Amstrad had been sufficiently party to any actual infringement which could render it to be an infringer and thus a joint tortfeasor. For such a tort to take hold, the incitement would have to be shown to have been directed to particular persons who could be identified or deemed identifiable at the date of the incitement. Consequently, no civil liability could arise if the incitement was merely directed towards the public at large.

The issue of criminal liability could not be conclusively established, and the court contented itself by conceding that it was the duty of Parliament, and not the Judges, "to provide new remedies for new wrongs."

In order to enable a plaintiff to sue for an injunction to restrain a criminal act, it is not deemed sufficient for him to merely show that the criminal act interferes with some property interest of which he is the owner.

In view of the fact that the copyright law provides for both civil and criminal liabilities, it could be inferred that since the act in question creates an obligation and enforces the performance in a specified manner, that performance cannot then be enforced in any other manner.

[44] Savola (2014), pp. 285–286.

[45] Ibid, p. 287.

[46] Key-Matuszak (2013), p. 440.

Where it so occurs that the copyright owners have no recourse to practical remedies as such against the actual infringers, then the courts are powerless to stop such activities and the Parliament alone is adapted best to deal with such situations (through the use of levies on the sale price of recording equipment, etc.).

4.1.3 The Diamond Rio MP3 Case: Space Shifting[47]

The third and final case which will be examined herein pertains to the digital revolution which, coupled with the Internet, led to the creation of a revolutionary novel method for distribution of music, thereby dealing more deadly blows to the music industry. In the late 1990s, an attempt was made once again by copyright owners in the United States to enjoin the manufacture, sale, marketing, and distribution of a portable entertainment system, namely the Rio MP3 player. The Rio was a small pocket-sized device with headphones. Its main feature was the ability to allow a user to download MP3 audio files from a computer and to listen to them at any place at his convenience.

The convenience of such a device cannot be understated. One just has to see it in the historical context to realize that the jump in recording technology from analog to digital had far-reaching benefits for the music listener. While earlier, if a person wanted to make a copy from a record or a compact disc, he could only use a cassette tape recorder. This was an analog-style recording, and it had its negative aspects/shortcomings. Consequently, every analog recording led to the intolerable situation that each successive generation of copies suffered progressively from high levels of degradation in the quality of the sound. On the other hand, digital copying does not show any degradation in the sound quality. This makes digital copying very attractive to music pirates who can make perfect copies of commercially prepared recordings, thereby infringing the copyrights subsisting therein.

This switch from analog to digital recording technology itself was of little consequence towards mass copying and distribution. This was because of the inherent limitations in the nature of the Internet itself in the early 1990s. Since the digital information contained within the average-sized music computer file tended to be excessively large, storing the same took an inordinate amount of space (requiring vast amounts of computer floppy discs) and downloading it from the Internet could take hours. This situation changed dramatically with the introduction of compression algorithm technology (including standard, nonproprietary, and freely available MPEG-1 Audio Layer 3, also known as MP3), which allowed an audio file to be easily made smaller by limiting its bandwidth.

Although this made downloading of music files from the Internet easy, it still meant that users could only listen to these songs by using speakers or headphones, while seated next to their computers. This was changed by the introduction of the

[47] Recording Industry Association of America v Diamond Multimedia Systems Inc 180 F. 3d 1072 (1999).

Rio device, whose main selling point was that it allowed for portability. Namely, the audio file could be downloaded from the Internet (or a compact disc player) onto the computer hard drive and then onto the Rio itself by plugging the Rio into the computer and with the aid of some special software known as the Rio Manager. It should be noted that the Rio device itself could not affect such a transfer and needed to be connected to a personal computer which had the Rio Manager software. The Rio could store vast amounts of sound files (up to 1 h of music and 16 h of spoken material such as eBooks, etc.), and with the addition of flash memory cards it was possible to store much more data content. The Rio could only be used for listening to the stored audio data via headphones but could not be used to make duplicates of any stored digital audio files. It could also not be used to transfer or upload such a file to any computer/device/Internet. However, by using a flash memory card, audio files could be removed from one Rio and played back in another.

The court examined the following pertinent points and drew far-reaching conclusions:

Although the predominant use of MP3 was stated to be trafficking of illegally downloaded audio recordings, especially by various pirate websites, leading to discouragement of legitimate purchases of audio recordings (losses alleged by the plaintiff were to the tune of over 300 million US Dollars), the court concluded that the legitimate business of sale and provision of free samples of audio files (including pre-recorded music) online by independent and wholly internet based record labels was growing rapidly and was according to some estimates worth more than a Billion US Dollars and therefore could not be ignored.

The Rio device was not required to meet the stringent requirements of the Audio Home Recording Act of 1992 with regard to the provisions for employment of Serial Copyright Management Systems that are designed to send, receive, and act upon information about the generation and copyright status of the files that it plays. This was because the Rio was held not to be a digital audio recording device (as defined by the Act) since it could only make copies from a computer hard drive and could not reproduce a digital music recording, either directly or from a transmission.

Even though there exist judicial precedents to the effect that straightforwardness of statutory command would bar any resort to legislative history, the court looked at both the plain language of the definitions of the Act and the legislative historical context (for interpretational purposes) and concluded that nothing could be seen as defining a digital musical recording as one which included songs fixed on computer hard drives. Notwithstanding the primary purposes of the recording function, it should be noted that a machine or a device is not to be considered a digital audio recording device even though it may have the technical capability to do so.

Limiting of the legislative exemption to computer programs meant that "any recording device could legally evade regulation by passing the music through the computer and ensuring that the MP3 file resided momentarily on the hard drive,"

and this was held to be indicative of legislative intent to create a specific loophole (perhaps in deference to the wishes of the powerful computer industry).[48]

The Rio facilitated personal, portable use for private, noncommercial purposes. This gave rise to the term "space-shifting" (of those files which already resided on a user's hard drive) and could be considered as "fair use."

Thus, it can be seen that the court followed the precedent laid down by the United States Supreme Court in the Sony case (regarding time shifting being fair use) by holding that space shifting was "paradigmatic noncommercial personal use."[49]

4.2 Possible Legal Solutions for Lawful 3D Printing

Looking forward, it could be said that the most promising analogies (among others) include architectural plans and blueprints, other technical drawings, computer programs, computer-generated works, and sculptures.[50] Different aspects of the existing categories ruin their suitability for 3D printing. For example, architectural plans may be similar to 3D printing in the sense that they exist first in the form of a CAD drawing and are later executed into physical objects.[51] Same goes for technical drawings, which usually consist of technical plans that normally would not be copyrightable but have been granted an exception under U.S. copyright law. Analogy to architectural plans or technical drawings would lead to a dual protection desired by the designer, meaning that both the CAD file and the 3D printed end result would be protected.[52] The main difference comes from the fact that the CAD file for an architectural building or a technical drawing includes guidelines for humans to interpret, while CAD files of a 3D printable object include information for a 3D printer to execute, and this underlying difference makes the analogy unsuitable.[53] 3D printed objects could easily fall under the copyrightable category of sculptures, but as some of them are not merely decorative and incorporate a utilitarian purpose, the objects sometimes fall out of the scope of copyright.

The analogy of 3D printing to computer programs can give different outputs to legal research. Firstly, the definition of computer programs could be used as analogous to 3D printing, and secondly, the anomaly of encompassing computer programs under copyright protection, as such, could give guidelines on how it would be possible to regulate and also encompass 3D printing technology under copyright protection. Computer programs under U.S. Copyright Act are defined as

[48] Ibid, p. 1079.

[49] Merges et al. (2012), p. 712.

[50] Dolinsky (2014), pp. 627–629.

[51] Osborn (2014), p. 829.

[52] Dolinsky (2014), pp. 629–631.

[53] Ibid.

"a set of statements or instructions to be used directly or indirectly in a computer program in order to bring about a certain result."[54] Some scholars have found the definition of computer programs perfectly compatible with CAD files because CAD files also "contain all the information to be used by a printer to print a three-dimensional model."[55] Why application of this analogy is not suitable is that a designer normally never writes the code of the CAD program but only uses the software to create a CAD design, which is not the equivalent of a software code written by a programmer.[56] Some CAD programs are even simplified to the extent that the designer only picks predesigned objects and aligns them according to his needs.[57] Thus, the copyrightability of computer programs extends to the software itself rather than to the work produced via the software.[58] Computer programs were protected as literary works, but the definition of a computer program in the EU computer programs directive was not very specific, in order to avoid the term becoming outdated and to allow for legal rules to follow the rapid development in technologies.[59] The regulation of computer programs was already established prior to them becoming more widespread for home use, and possibly the legal regulation played a part in the success and innovation that followed computer programs. Though many categories could suffice to act as an equivalent to either CAD files or 3D printed objects, none is capable of simultaneously encompassing both the digital and physical features accompanying the technology. Due to the complex nature of the whole 3D printing technology and the different steps from CAD file to the actual 3D printing process, it could be reasonable to try and regulate it more specifically, by setting up a legal framework to improve legal clarity.

CAD files and 3D printed objects are a unique form of expression to copyright law and do not completely comply or fall under any of the existing categories of copyrightable subject matter and due to their complexity pose new issues and questions about the suitability under copyright protection regarding the functionalities of CAD file and the 3D printed end result. For these reasons, it has been proposed by scholars that it would be reasonable to establish a sui generis copyright-like protection for 3D objects. Whenever a novel technology accompanied with economic benefits emerges, policy makers need to make considerations in order to provide suitable legal framework for the new technologies to operate,

[54] 17 U.S.C. section 101.

[55] Osborn (2014), p. 824.

[56] Ibid, p. 829.

[57] Ibid.

[58] Dolinsky (2014), pp. 637–639.

[59] Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs. OJ L 111/16, 5.5.2009, recital (7). The term "computer program" has been somewhat defined for the purpose of the directive under the preamble, and it "/../shall include programs in any form, including those which are incorporated into hardware. This term also includes preparatory design work leading to the development of a computer program provided that the nature of the preparatory work is such that a computer program can result from it at a later stage."

because the protection of such works will have impact on the technological development.[60] In the EU, sui generis protection has been granted for databases with the database directive.[61] Though the originality and suitability of many databases under copyrightable subject matter is doubtful, the objectives of granting databases a sui generis protection under copyrights include the substantial investments required from the maker of the database in order to create the database and the fact that copyrights remain the most appropriate form of IP protection for authors of databases.[62] So far, databases, which do not qualify for traditional copyright protection, are the only exception of works to be granted sui generis protection under EU copyright law, but it has been previously suggested by scholars that computer programs should have also been protected with a sui generis right. Computer software falls somewhere in between copyright and patent rights, and it has been declared that copyrights provide insufficient protection, while patent law is too restrictive for innovation and development of the technology.[63] In practice, protecting computer programs as literary works within the meaning of the Berne Convention can already be seen as implementing a sui generis right because the traditional copyright rules have been widened and altered to comply with the distinctive technological characteristics of computer programs.[64] Taking into account the fact that no such subject matter has previously existed in the realm of copyright protection and that it incorporates digital and physical aspects both seeking copyright protection, the sui generis proposals by scholars for 3D printed objects is not an entirely unexpected line of thought. The sui generis right that Rideout proposes for 3D printing technology is to establish a copyright-like protection for even those 3D printed objects that incorporate a useful article and, as previously determined, would thus fall out of the scope of copyright. Rideout generates the idea on the basis of sui generis right granted for vessel hulls under U.S. copyright law, which resembles industrial design protection and applies to the appearance and utilitarian function of the vessel hull.[65] Thus, he proposes that the necessary practice of protecting works with a sui generis right under the scope of copyright exists and it could be easily broadened to encompass 3D printed objects as well.[66] Creating a sui generis protection for 3D printing technology would merely constitute a method of encompassing all 3D printed objects, as such, under copyright protection. It would make it very convenient for designers, as there will be no reason for obtaining industrial design protection or trademark protection to pursue their intellectual property protection, because copyrights for

[60] Mylly (2009), p. 880.

[61] Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases OJ L 077, 27.03.1996.

[62] Ibid, recital (5), (7).

[63] Toeniskoetter (2005), p. 76.

[64] Mylly (2009), p. 880.

[65] Rideout (2011), p. 175.

[66] Ibid.

3D printed objects would exist automatically. This solution could possibly decrease destructive effects of regulation to 3D printing industry, as the protection of works can create a higher incentive for designers to create and share their designs. On the other hand, it could also have a negative effect on the traditional intellectual property regulations in place because it is capable of creating multiple layers of protection by different forms of IP; for example, the end result can be simultaneously protected by copyright and design right, which can end in overprotecting of works, which is also unreasonable and not the purpose of setting the sui generis protection.

Michael Weinberg and other scholars have expressed concerns that such sui generis copyright-like protection for functional objects will create a patent-like protection, without the novelty requirement and strict period of protection, which is usually granted for 20 years.[67] Patents are meant to protect useful creations and are rewarded to inventions, which are novel and have inventive step. The application process is complicated and costly, which is why they are hard to ascertain. 3D printing can bring forth problems for 3D enthusiasts, even when they independently create the design for an infringing object, which is not the case with copyrights.[68] On the other hand, taking into account the desktop 3D printer quality and materials currently available, there might not be many patented objects that could be executed through 3D printing.[69]

Copyrights and design rights are very similar to one another, as the object of protection for both is the visual appearance of a work. In the EU, a great emphasis is put on highlighting the importance of design and to support that, a harmonized Community Design system is established with Council Regulation 6/2002. The harmonization is carried from the idea of creating a designer-friendly environment, in which innovation of, development of, and investments into new products are encouraged.[70] In case of copyrights and design rights, one does not exclude the other, and they can exist cumulatively for a work. Design is a key element for being successful in business and competition—it helps for the product to stand out in the variety of others. 3D printing is especially beneficial for designing new test products, as it helps to make the design from digital to physical in a matter of hours, simplifying the creation of test products and making the production process and entering to market much faster than it has been before.[71] At the same time, the digital era is a stepping stone for designers, who now have to think about protecting their works more than ever prior to publishing any of their designs and making them vulnerable for intellectual property infringements, which can be utilized into a product in a very small time frame. Design law and copyright law are closely related when it comes to 3D printing, mainly for the reason that if and when the

[67] Ibid.

[68] Doherty (2012), p. 359.

[69] Bradshaw et al. (2010), pp. 26–27.

[70] Council Regulation (EC) No 6/2002 of 12 December 2001 on Community designs, recital (7).

[71] Lewis (2014), pp. 315–316.

range of materials for 3D printing escalates, it will enable printing of many different utilitarian works, such as leather shoes, clothing, and so on, which are generally excluded from the protection of copyrights, due to their utilitarian nature, and are the reason why design law was generated. In general, it is possible that the CAD files could acquire copyright protection, while 3D printed objects which are on the borderline of copyrights, but suitable for design protection, will fall under the scope of design protection. In the case of adequate design regulation, it would be a clear and good solution, which would eliminate the need to expand copyright law to functional objects and would help to avoid duplicate layers of IP protection for 3D printed objects.

5 A Possible Technical Solution for Effective and Lawful 3D Printing Using Secured Streaming of 3D Designs

The whole value chain from idea to a physical object should be secured, as the design could be potentially compromised at any step. It is not just a new type of streaming; it is a comprehensive set of tools combined on an ultimate cloud security platform, which includes 3D printing copyright protection available to use at every step through the whole value chain, secured streaming to 3D printers and between secured cloud servers, detective and protective controls allowing to detect intruder even before he can compromise secured stream to 3D printers. Secured streaming of 3D designs is built on the philosophy cloud versus hacker—a human being with a cloud for hacking. In the age of cloud computing, intrusion to almost any system is a matter of time and computing power. One important integral part of the solution for 3D design cloud storage and streaming is the ability of the system to set time-based limitations. Simplified IP-secured delivery process is shown in Fig. 2.

Majority of 3D printers that are available on the market are not network enabled; most of 3D printers utilize USB. Industrial and professional printers do have network connectivity but in most of the cases for file transfer only; the printing process and settings of the machine happen inside the machine through the touch screen interface or special software. In Fig. 3 there are three different approaches how secured stream can reach the 3D printer. The preferable approach is embedded cloud client, which is also a decrypting module for the secured streaming; this

Fig. 2 Simplified IP secured delivery process

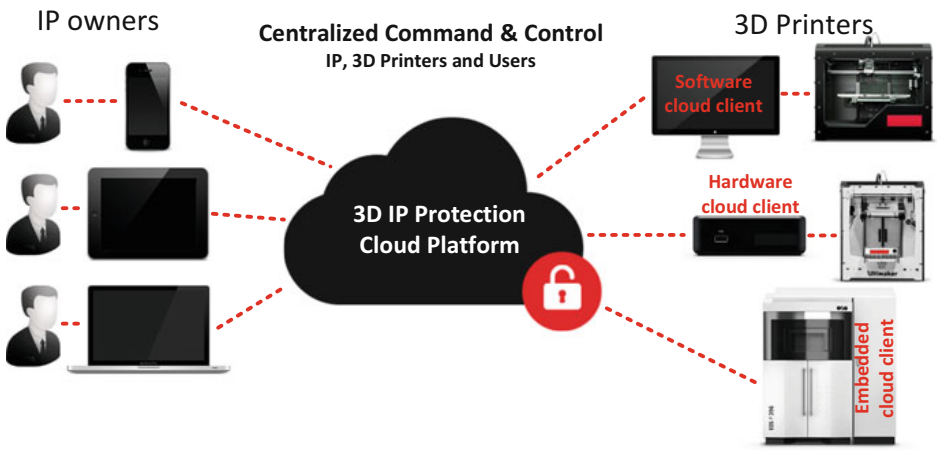


Fig. 3 Secured streaming cloud client types and methods of connection

approach is the most secure, and cloud client is implemented on a hardware chip, which is installed as a part of 3D printer main board. This approach is valid for newly produced printers or disassembly, and change to hardware is needed. Hardware cloud client is the second preferable solution from a security point of view, easy to implement too; the requirement is to keep a decryption box as close as possible to the printer, as USB connection still could be vulnerable. Hybrid solution is also possible; when decryption box decrypts the stream and encrypts it for USB-secured transfer, this type of encryption needs less code on the printer side and could be just included into 3D printer firmware. For example, 3D printer open source firmware Marlin could be changed in a way that it decrypts secured USB connection.

In Fig. 4, you can see the conceptual diagram of high-level cloud architecture which gives a general idea on how secured storage and streaming is built. It consists of four main components: Web, File Segments, Key and Streaming Cloud Module, and one optional (smart card). To mitigate intrusion, the information is segmented within the cloud platform.

It is important to understand that File Segments machines on the left and Key machines on the right are autonomous and proactive, which means there is no way to query or send a command to them; they behave according to their internal rules, monitor Web machines on the top and Streaming machines on the bottom of the figure above, make decisions whether it is secure and as regards the right moment to transfer any information. So basically the intruder has to analyze for a long time the behavior of File Segments and Key machines to get any idea how exactly they are operating and what the possible vulnerabilities are; by that time, the intruder will be detected and measures will be taken.

The whole process starts from the Web cloud module. Web server receives a file. File Segments machine is monitoring the web server for new files. As soon as there is a new file, it is taken by the File Segments machine. Key machine also monitors

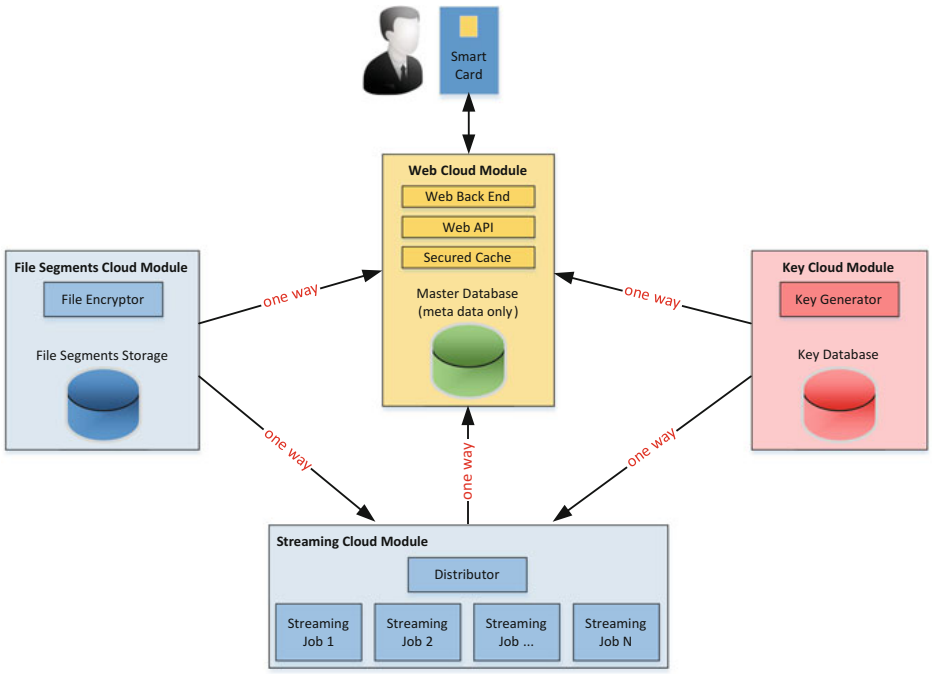


Fig. 4 Secured storage and streaming high level cloud architecture conceptual diagram

the Web machine and is ready to generate a set of asymmetric key pairs. Usually there are thousands of keys generated. Private keys are kept on the Key machines and never exposed until streaming process. A File Segments machine collects the keys, splits the file into thousands of pieces, and encrypts each segment with its own key. File Segments are kept on File Segments machine until streaming process starts. There is no way to get all segments at the same time slot. File Segments module has a special type of storage which will physically allow to get more segments than a 3D printer is physically able to print. The next step is secured streaming process. User sends a command to print a design. Web module stores a request in the queue. All three servers (File Segments, Key, Streaming) analyze the metadata on the web machine and make a decision to start the streaming process. Streaming machines create a temporary virtual machine or a container for the moment of streaming, which will be deleted right after the streaming process. Streamer communicates and gets ready the 3D printer to receive the stream. File Segments issue a first segment of a file and sends it to the streamer virtual machine created for that exact job. Key module waits till all the servers are ready and sends over a private key for that exact segment. Streamer receives a key, decrypts a segment, and encrypts it for streaming. Streaming encryption works as all-the-time-changing hash table.

Thus, there is no single point of failure in case of intrusion, and until more than 1 server type is compromised the system is not vulnerable. In production system, each type of servers is actually a cloud by itself, so imagine 10–100 virtual

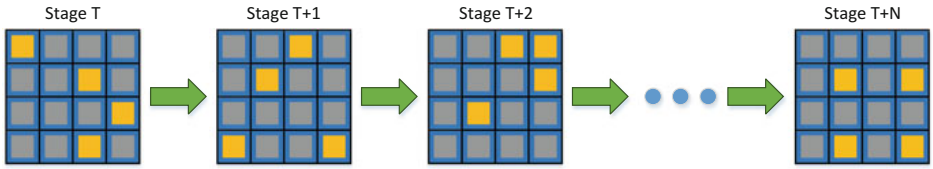


Fig. 5 Live matrix concept, changing state millions of times per short time frame

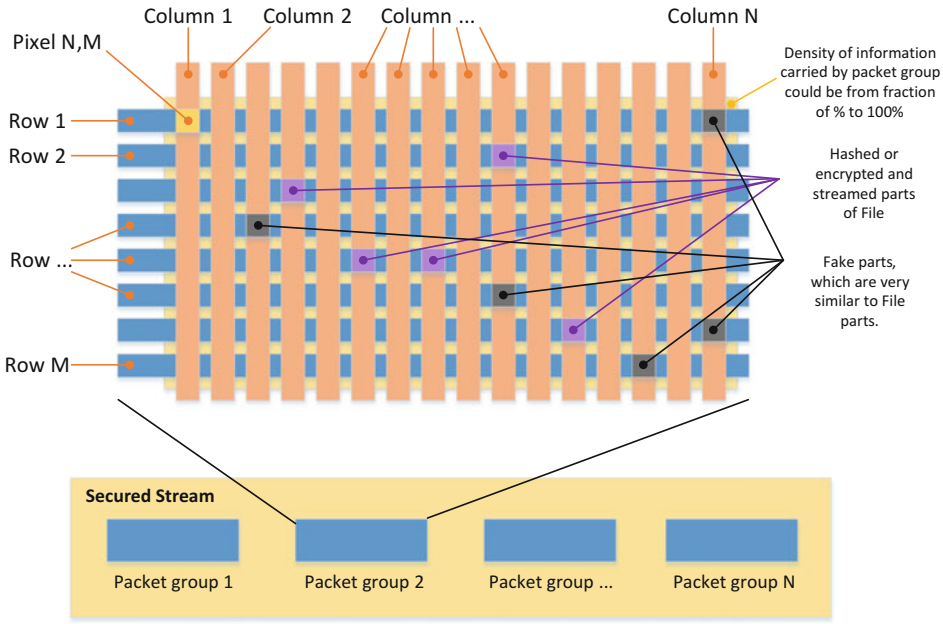


Fig. 6 Simplified principle of a live matrix concept

machines in the place of each block on the figure below. And the data is segmented within this sub-cloud, which makes it even harder to compromise.

The data is kept in so-called live matrixes, a hash-table-like structure, which change their state millions of times per short time slot. A live matrix is calculated on a server and in decryption module. There is no physical possibility to keep more than 2–3 versions of these structures in decryption module; basically, it makes impossible to decrypt parts of the stream half a minute later. If hacker will record the whole stream going to 3D printer, half a minute later even decryption module could not decrypt it, so it is not possible to "replay" the stream, as it is sometimes possible in case of media streams. In Fig. 5 is a conceptual diagram showing so-called live matrix life cycle.

In Fig. 6, you can see a detailed view on the live matrix structure and how it works. The file is split into thousands of splits, each split is split into many parts, every part is hashed and is located at its own place in the matrix, and the matrix is changing its state all the time by rehashing the values. The same function runs in a

decryption box of a 3D printer, and when a new hash is coming, it is being looked up in the live matrix. Fake parts could be added to make cracking more complicated.

6 Applications of Secured Streaming in Real Businesses

3D design marketplaces use secured streaming technology to protect IP of designers. Now it is possible not only to protect 3D designs on the way to 3D printer but also to provide a possibility to sell one-time-print licenses, which allow end users to print the desired object only once. In case of a technical problem, user is allowed to print one more time, but in order to do that, he needs to make a picture of a failed object and send it to a support desk, then another one time license is granted. This market is just evolving, but already today there are good examples like Pinshape, which serves as a marketplace for downloading or streaming of 3D printable models.[72] A typical secured streaming and 3D-copyright-protection-enabled 3D marketplace business process is shown in Fig. 7.

Many companies will change their business models because of advancing 3D printing technology. For example, LEGO—"Will 3D printing turn Lego into an intellectual property publisher?"[73] There is a true story—a child once a week was sending to LEGO HQ a 3D printed part; initially it was rough and not fitting well,

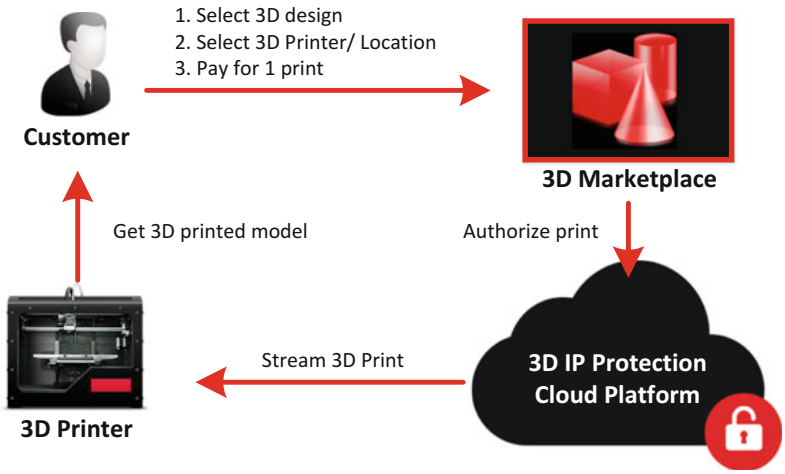


Fig. 7 Typical secured streaming and 3D copyright protection enabled 3D marketplace business process

[72] See more from: www.pinshape.com.

[73] Levine (2014). Will 3D Printing Turn Lego Into an Intellectual Property Publisher? Available at: http://venturebeat.com/2014/03/03/will-3d-printing-turn-lego-into-an-intellectual-property-publisher/ (accessed 20.08.2015).

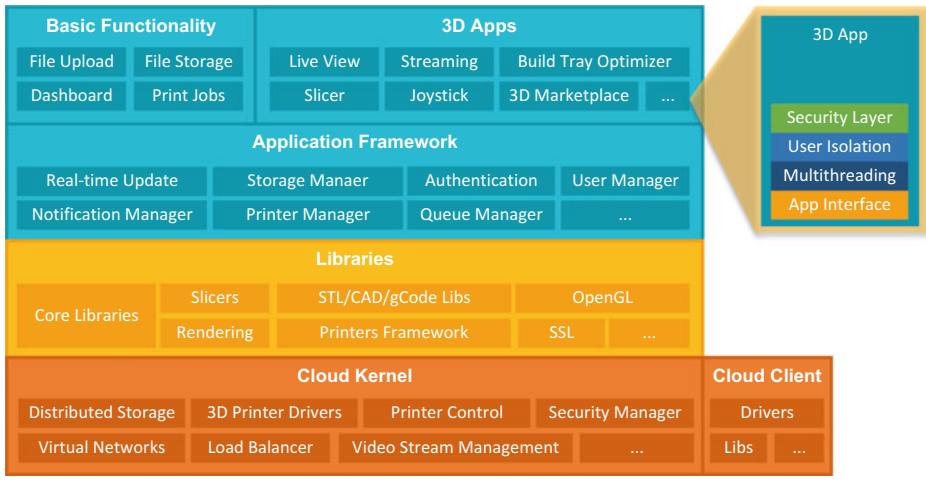


Fig. 8 3DprinterOS—general architecture of open operating system for 3D printers

and Lego executives did not take it seriously; however, half a year later, a teenager could so well fine-tune printing settings of his 3D printer that the plastic part fitted perfectly. The same way when 15 years ago many industries switched to media streaming through the Internet instead of CD/DVDs, now we are at the edge of next revolution when 3D printing will change the way companies operate. So maybe in 3–5 years people would buy a license to print a Lego set, instead of buying one.

Due to the need for end-to-end 3D printing, copyright solutions to secure the whole value chain cloud software platforms evolve. An interesting example is 3DprinterOS represented in Fig. 8—a cloud-based open operating system for 3D printers.[74] It is like an Android for 3D printers but runs its apps in the cloud. Compared to usual AppStore or Google Play applications, one of the essential things for every app developer is to implement the security layer, which is compatible with secured streaming of 3D designs. This has the potential to secure the IP of the whole ecosystem and 3D printing value chain.

There are many more examples of famous designers, design bureaus, 3D print hubs, 3D printer manufacturers, 3D marketplaces, schools and universities, 3D print shops, production and prototyping companies that every day in their business process already use secured streaming of 3D designs to protect their copyrights and prevent the stealing of their IP.

[74] See more from: www.3dprinteros.com.

7 Conclusion

Thus, we have seen that the existing measures for enforcing copyright protection need to be reviewed in order to comply with the complex nature of 3D printing technology. Based on an analysis of existing copyright regulation in relation to 3D printing, a conclusion can be made that the existing copyright regulations are not capable of encompassing the entire process of 3D printing. Some analogies are compatible at parts, but none is fully suitable, and it could even be possible that CAD files can be protected under copyrights and the physical 3D printed objects under design rights. In terms of existing regulation, it definitely needs to be reviewed before introducing the subject matter of 3D printing under copyright regulation. Because no such subject matter has been regulated before, the different alternatives also need to be carefully considered and reviewed; the possible outcomes of implementing regulative measures should be evaluated, to achieve an efficient regulative solution for 3D printing. Perhaps to refrain from interfering with the innovation and technological development, the industry should rather be left unregulated for as long as possible, to allow for it to mature and develop. Applying strict DRM and copyright protection cumulatively can lead to overregulating the industry and might end in decreasing innovation, which is why a DRM-like solution should consider the industry-specific characteristics to provide a suitable solution. The world of 3D printing is exciting and is capable of offering endless opportunities to different fields of use, if we only allow. The authors have brought forward the legal issues and have attempted to describe a unique technical solution—secured streaming which solves or at least partially solves the problem of copyrights in 3D printing. The proposed solution provides a possibility for a copyright owner to limit the number of 3D prints. He can specify the number of copies that are allowed for the manufacturer or an end user to produce. Moreover secured streaming has detective and protective controls to detect information system compromises and to stop streaming of 3D designs to 3D printers.

References

Akester P (2005) Copyright and the P2P challenge. Eur Intellect Prop Rev 27(3):106–112

Bradshaw S, Bowyer A, Haufe P (2010) The intellectual property implications of low-cost 3D printing. ScriptEd 7(1):7–8

Daly M (2007) Life after Grokster: analysis of US and European approaches to file-sharing. Eur Intellect Prop Rev 29(8):319–324

Dasari H (2013) Assessing copyright protection and infringement issues involved with 3D printing and scanning. Am Intellect Prop Law Assoc Q J 41:279

Doherty D (2012) Downloading infringement: patent law as a roadblock to the 3D printing revolution. Harv J Law Technol 26:358

Dolinsky K (2014) CAD's cradle: untangling copyrightability, derivative works, and fair use in 3D printing. Washington Lee Law Rev 71:629–631

Garnett KM, Davies G, Harbottle G (2005) Copinger and Skone James on Copyright. Sweet & Maxwell, London
Giblin R (2012) Stranded in the technological dark ages: implications of the Full Federal Court's decision in NRL v Optus. Eur Intellect Prop Rev 34(9):632–641
Haque H (2008) Is the time ripe for another exclusive right? Eur Intellect Prop Rev 30(9):371–378
Howells JAJ (2014) The intellectual property right implications of consumer 3D printing. Available at: http://pure.au.dk/portal-asb-student/files/71036699/The_Intellectual_Property_Right_Implications_of_Consumer_3D_Printing_Final.pdf (accessed: 06.08.2015), p 13
Karapapa S (2011) Padawan v SGAE: a right to private copy? Eur Intellect Prop Rev 33 (4):252–259
Key-Matuszak P (2013) Time-shifting after NRL v Optus: a need for amendments. Eur Intellect Prop Rev 35(8):439–444
Khaosaeng K (2014) Wands, sandals and the wind: creativity as a copyright exception. Eur Intellect Prop Rev 36(4):238–249
Lewis A (2014) The legality of 3D printing: how technology is moving faster than the law. Tulane J Technol Intellect Prop 17:315–316
Merges RP, Menell PS, Lemley MA (2012) Intellectual property in the new technological age. Wolters Kluwer Law and Business, Aspen Casebook Series, New York
Monotti AL (2013) Liability for joint infringement of a method patent under Australian law. Eur Intellect Prop Rev 35(6):318–326
Mylly UM (2009) Harmonizing copyright rules for computer program interface protection. Univ Louisville Law Rev 48
Nyman-Metcalf N, Dutt PK, Chochia A (2014) The freedom to conduct business and the right to property: the EU technology transfer block exemption regulation and the relationship between intellectual property and competition law. In: Kerikmae T (ed) Protecting human rights in the EU. Springer, Berlin, pp 37–70
Osborn LS (2014) Of PhDs, pirates and the public: three-dimensional printing technology and the arts. Texas A&M Law Rev 1
Raval MI (2012) Game over for mod chips? The aftermath of Sony v Stevens and the Australian-US Free Trade Agreement. Eur Intellect Prop Rev 34(2):95–107
Rideout B (2011) Printing the impossible triangle: the copyright implications of three-dimensional printing. J Bus Entrep Law 5:167–168
Savola P (2014) Blocking injunctions and website operators' liability for copyright infringement for user-generated links. Eur Intellect Prop Rev 36(5):279–288
Stahl H (2013) 3D printing–risks and opportunities. Öko-Institut e.V. Institute for Applied Ecology, pp 3–4
Toeniskoetter SB (2005) Protection of software intellectual property in Europe: an alternative sui generis approach. Intellect Prop Law Bull 10
Twomey P (2014) A new dimension to intellectual property infringement: an evaluation of the intellectual property issues associated with 3D printing. Trinity Coll Law Rev 17:33
Weinberg M (2013) What's the deal with copyright and 3D printing?. White paper from Public Knowledge's Institute for Emerging Innovation, p 1
Xiaoxiang Shi S (2012) Time shifting in a networked digital world: Optus TV Now and copyright in the cloud. Eur Intellect Prop Rev 34(8):519–533
Yan M (2012) The law surrounding the facilitation of online copyright infringement. Eur Intellect Prop Rev 34(2):122–126

Others

Sony Corporation of America v Universal City Studios, Inc., 464 U.S. 417 (1984).

CBS Songs Ltd v Amstrad Consumer Electronics Plc [1988] A.C. 1013
Recording Industry Association of America v Diamond Multimedia Systems Inc 180 F. 3d 1072 (1999)
Metro-Golwyn-Mayer Studios Inc. v. Grokster, Ltd. Supreme Court of the United States 545 U.S. 913 (2005).
Copyright Act 1956 (UK)
Copyright act 1976, 17 U.S.C.
Digital Millennium Copyright Act, 1988
Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases OJ L 077, 27.03.1996.
Charter of Fundamental Rights of the European Union (2000/C 364/01)
Council Regulation (EC) No 6/2002 of 12 December 2001 on Community designs
Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs. OJ L 111/16, 5.5.2009.
Levine, B. (2014). Will 3D Printing Turn Lego Into an Intellectual Property Publisher? Available at: http://venturebeat.com/2014/03/03/will-3d-printing-turn-lego-into-an-intellectual-property-publisher/ (accessed 20.08.2015)
Gartner, (2013). Press release: Gartner Reveals Top Predictions for IT Organizations and Users for 2014 and Beyond. Available at: http://www.gartner.com/newsroom/id/2603215 (accessed 20.08.2015)
New Balance (2013). Press release: New Balance Pushes the Limits of Innovation with 3D Printing. Available at: http://www.newbalance.com/press-releases/id/press_2013_New_Balance_Pushes_Limits_of_Innovation_with_3D_Printing.html (accessed 20.08.2015)
Walters, R. (2012). The Pirate Bay Declares 3D Printed "Physibles" as the Next Frontier of Piracy. Available at: http://www.extremetech.com/electronics/115185-the-pirate-bay-declares-3d-printed-physibles-as-the-next-frontier-of-piracy (accessed 06.08.2015).

Appendix 5

V

K. Isbjornssund and A. Vedeshin. Method and system for enforcing 3d restricted rights in a rapid manufacturing and prototyping environment, Feb. 27 2014. US Patent App. 13/973,816

US 20140058959A1

(19) **United States**

(12) **Patent Application Publication**

ISBJORNSSUND et al.

(10) **Pub. No.: US 2014/0058959 A1**

(43) **Pub. Date: Feb. 27, 2014**

(54) **METHOD AND SYSTEM FOR ENFORCING 3D RESTRICTED RIGHTS IN A RAPID MANUFACTURING AND PROTOTYPING ENVIRONMENT**

(71) Applicants: **Kimmo ISBJORNSSUND**, Tallinn (EE); **Anton VEDESHIN**, Tallinn (EE)

(72) Inventors: **Kimmo ISBJORNSSUND**, Tallinn (EE); **Anton VEDESHIN**, Tallinn (EE)

(21) Appl. No.: **13/973,816**

(22) Filed: **Aug. 22, 2013**

(30) **Foreign Application Priority Data**

Aug. 21, 2012 (EP) EP12181430.5

Publication Classification

(51) **Int. Cl.**

G06Q 50/18 (2006.01)

G06Q 10/10 (2006.01)

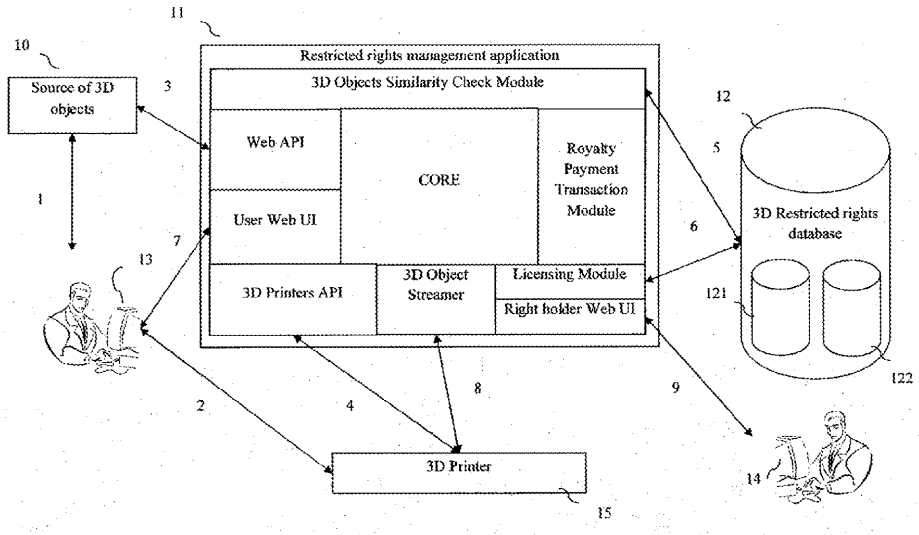
(52) **U.S. Cl.**

CPC ***G06Q 50/184*** (2013.01); ***G06Q 10/10*** (2013.01)

USPC **705/310**

(57) **ABSTRACT**

A system and method of enforcing 3D restricted rights in a rapid manufacturing and prototyping environment may include, in response to receiving a 3D object data representative of a 3D object, performing at least one function on the 3D object data to determine a parameter set for each respective function. Business rule(s) may be applied to each parameter set for each respective function. At least one algorithm may be performed to determine whether at least a portion of the 3D object matches a rights restricted 3D object. In response to determining that at least a portion of the 3D object matches a restricted rights 3D object, an action may be caused to be taken, otherwise, in response to determining that at least a portion of the 3D object does not match a restricted rights 3D object, the 3D object may be enabled to be rapid manufactured or prototyped.



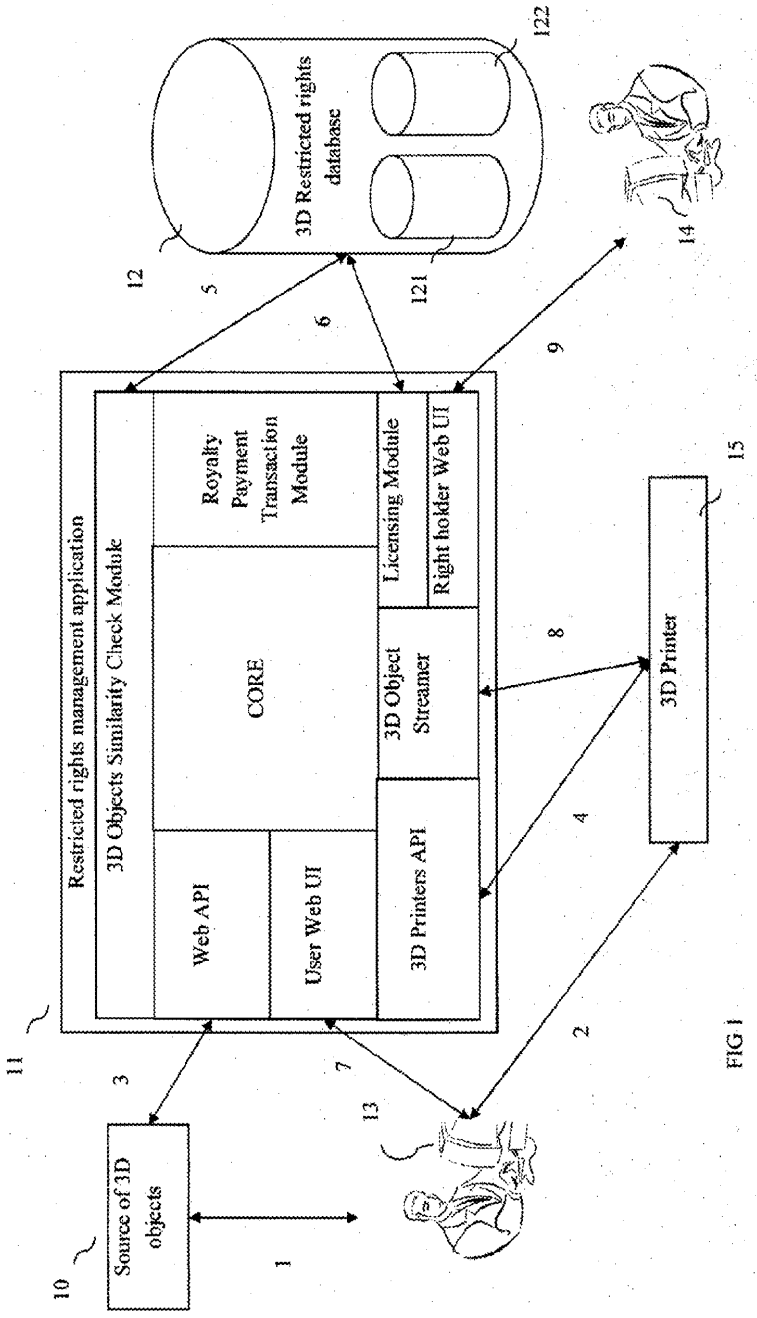

Source of 3D objects
10
1
3
11
Restricted rights management application
3D Objects Similarity Check Module
Web API
User Web UI
CORE
Royalty Payment Transaction Module
3D Printers API
3D Object Streamer
Licensing Module
Right holder Web UI
7
13
2
4
8
3D Printer
15
12
5
6
3D Restricted rights database
121
122
9
14

FIG 1

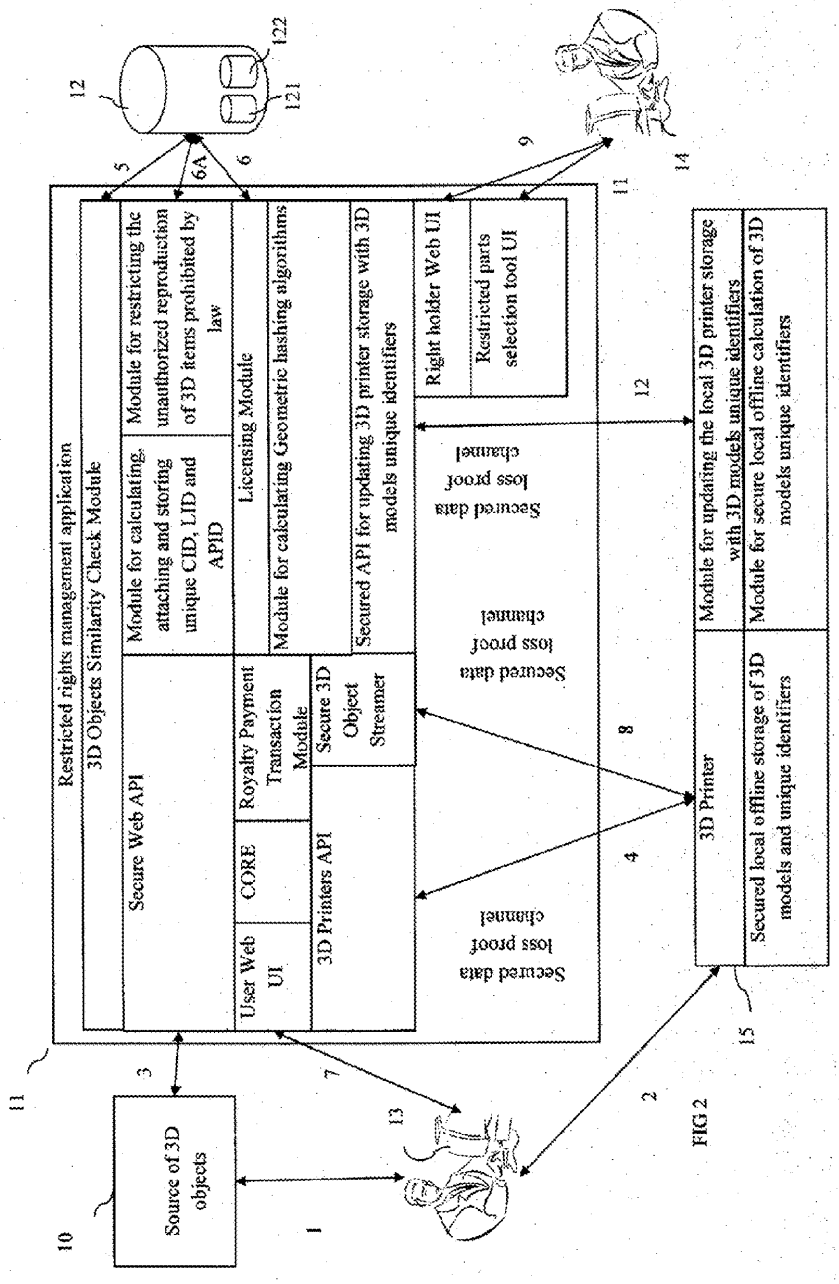

Source of 3D objects
10
1
3
11
Restricted rights management application
3D Objects Similarity Check Module
Secure Web API
User Web UI
CORE
3D Printers API
Royalty Payment Transaction Module
Secure 3D Object Streamer
Module for calculating, attaching and storing unique CID, LID and APID
Module for restricting the unauthorized reproduction of 3D items prohibited by law
Licensing Module
Module for calculating Geometric hashing algorithms
Secured API for updating 3D printer storage with 3D models unique identifiers
Right holder Web UI
Restricted parts selection tool UI
Secured data loss proof channel
Secured data loss proof channel
Secured data loss proof channel
12
121
122
5
6
64
7
8
4
2
9
11
12
13
14
15
3D Printer
Secured local offline storage of 3D models and unique identifiers
Module for updating the local 3D printer storage with 3D models unique identifiers
Module for secure local offline calculation of 3D models unique identifiers

FIG 2

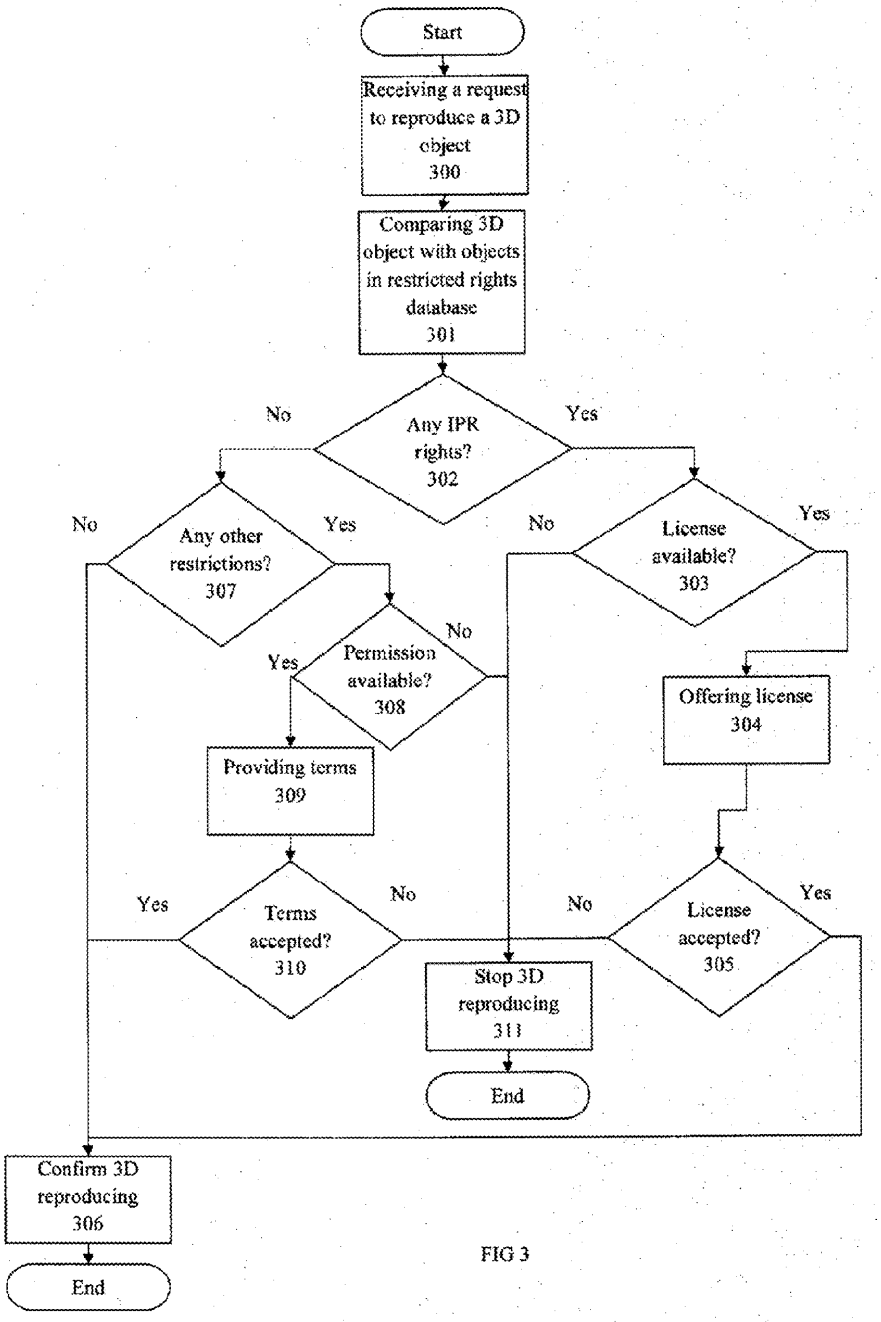

Start
Receiving a request to reproduce a 3D object 300
Comparing 3D object with objects in restricted rights database 301
Any IPR rights? 302
No
Yes
Any other restrictions? 307
No
Yes
License available? 303
No
Yes
Permission available? 308
Yes
No
Providing terms 309
Offering license 304
Terms accepted? 310
Yes
No
License accepted? 305
No
Yes
Stop 3D reproducing 311
End
Confirm 3D reproducing 306
End

FIG 3

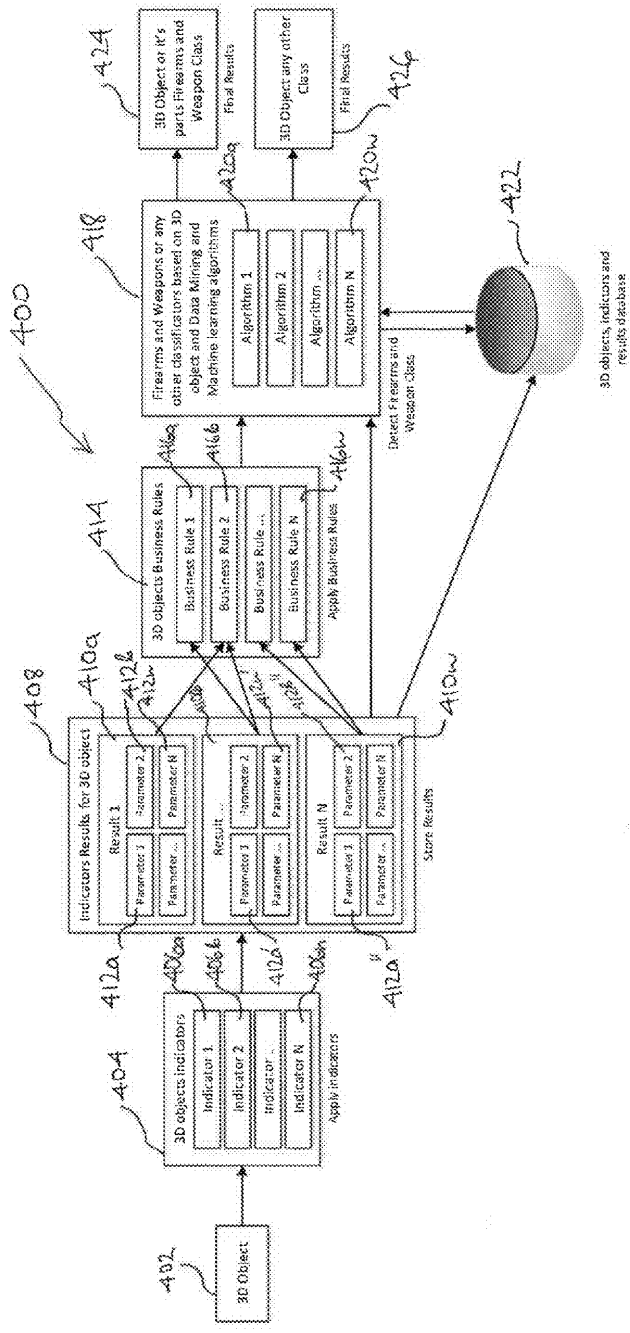

400
402
3D Object
404
3D objects Indicators
Indicator 1
Indicator 2
Indicator ...
Indicator N
Apply Indicators
406a
406b
406n
408
Indicators Results for 3D object
Result 1
Result ...
Result N
Parameter 1
Parameter 2
Parameter ...
Parameter N
410a
410n
412a
412b
412n
414
3D objects Business Rules
Business Rule 1
Business Rule 2
Business Rule ...
Business Rule N
Apply Business Rules
416a
416b
416n
418
Firearms and Weapons or any other classifications based on 3D object and Data Mining and Machine learning algorithms
Algorithm 1
Algorithm 2
Algorithm ...
Algorithm N
420a
420n
Detect Firearms and Weapon class
422
3D objects, indicators and results database
Store Results
424
3D Object or it's parts Firearms and Weapon Class
Final Results
426
3D Object any other Class
Final Results

FIG.4

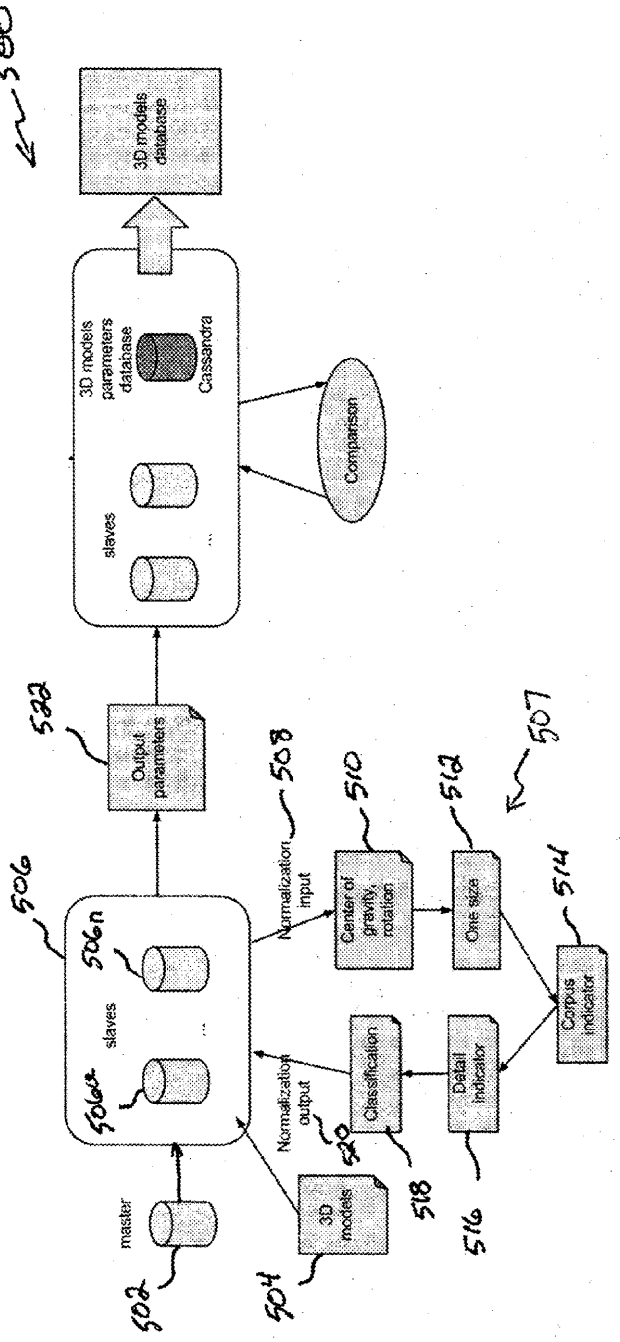

500
3D models database
3D models parameters database
Cassandra
slaves
Comparison
522
Output parameters
506
506n
slaves
506a
master
502
3D models
504
Normalization input
Normalization output
Classification
520
518
Detail indicator
516
508
Center of gravity, rotation
510
One size
512
507
Corpus indicator
514

FIG. 5

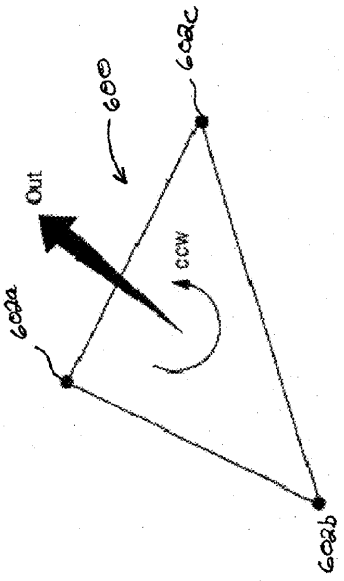


FIG. 6

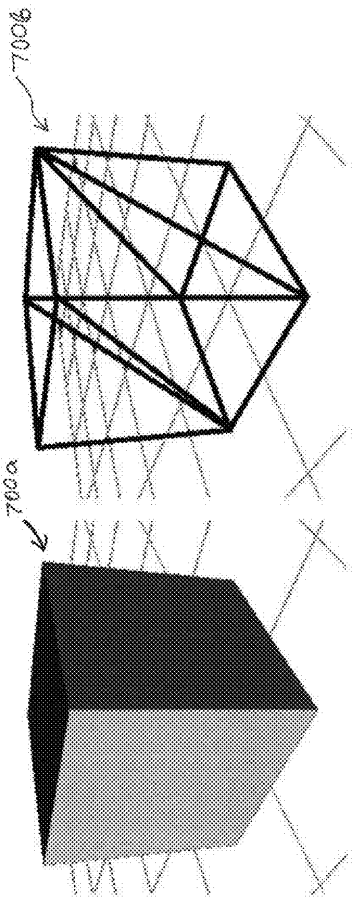


FIG. 7

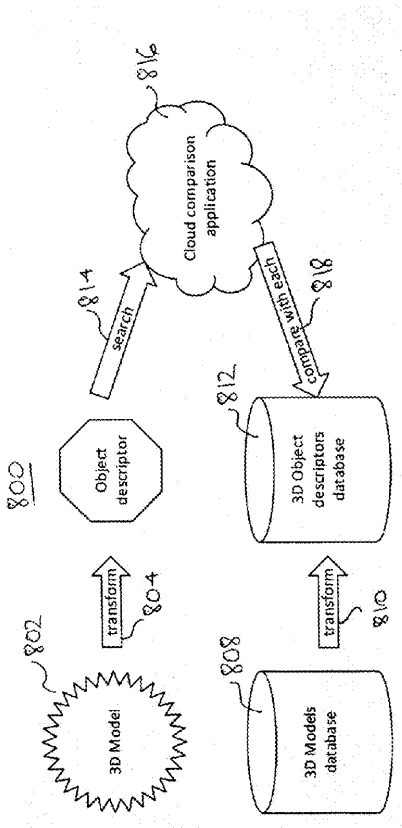

800
802
3D Model
804
transform
Object descriptor
808
3D Models database
810
transform
812
3D Object descriptors database
814
search
816
Cloud comparison application
818
compare with each

FIG.8

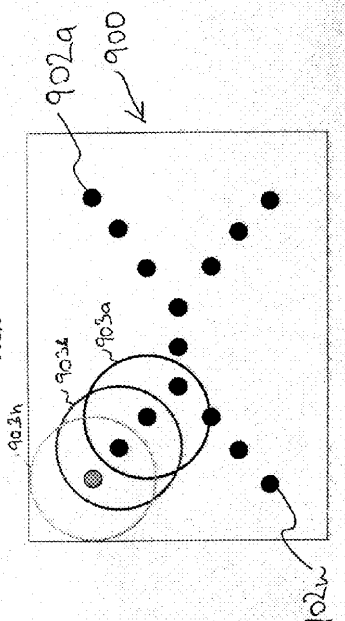

900
902a
902n
903a
903b
903n

FIG.9

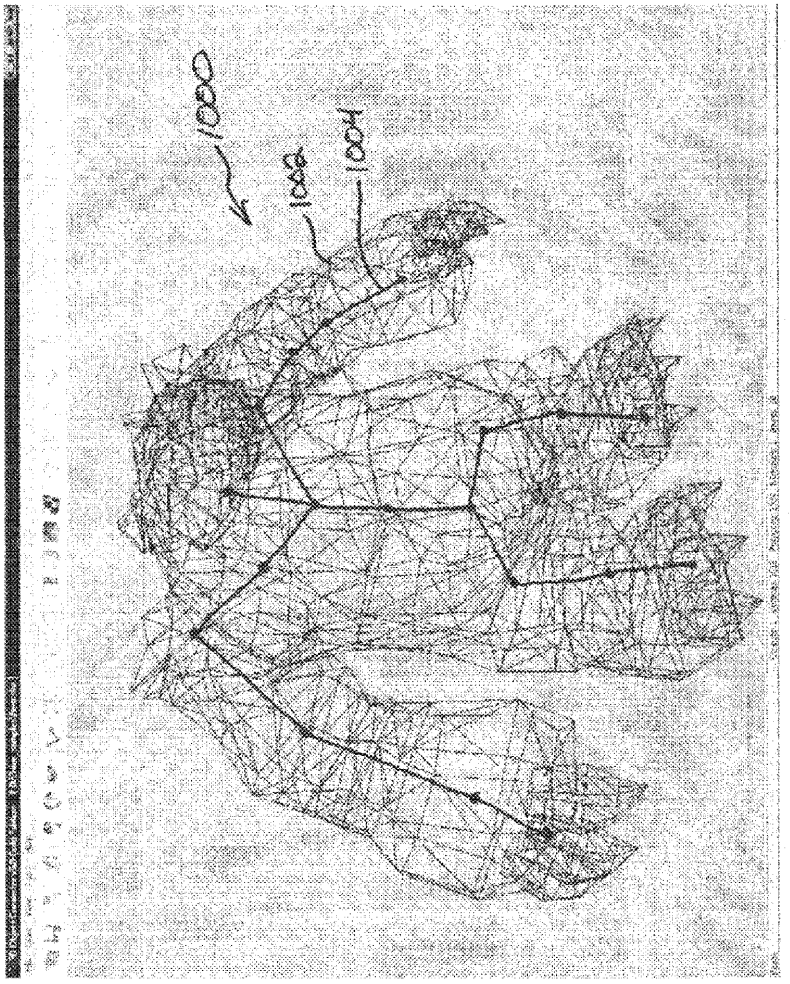


FIG. 10

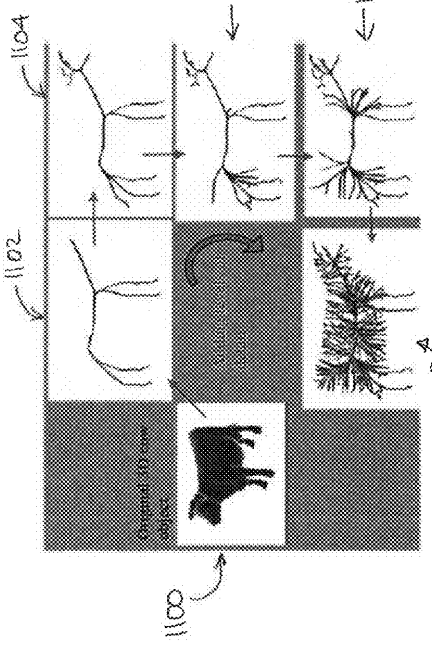

1100
1102
1104
1106
1108
1110
1111

FIG. 11

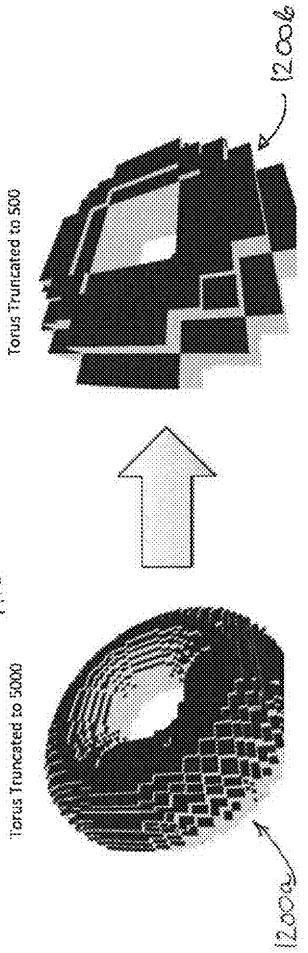

Torus Truncated to 5000
Torus Truncated to 500
12002
12006

FIG. 12

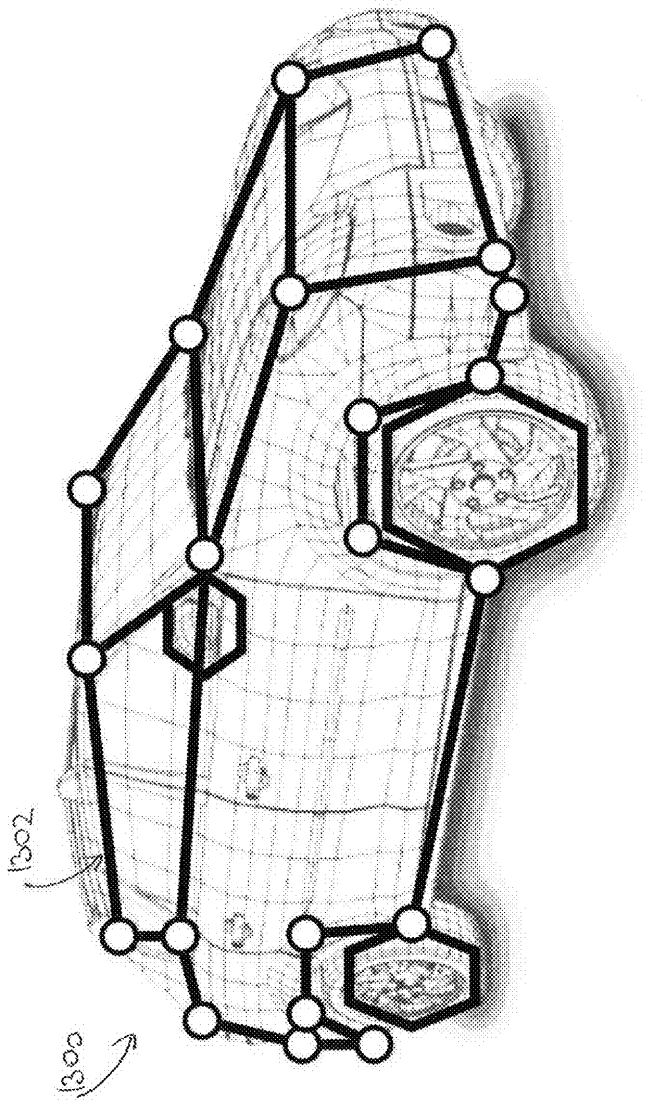


FIG.13

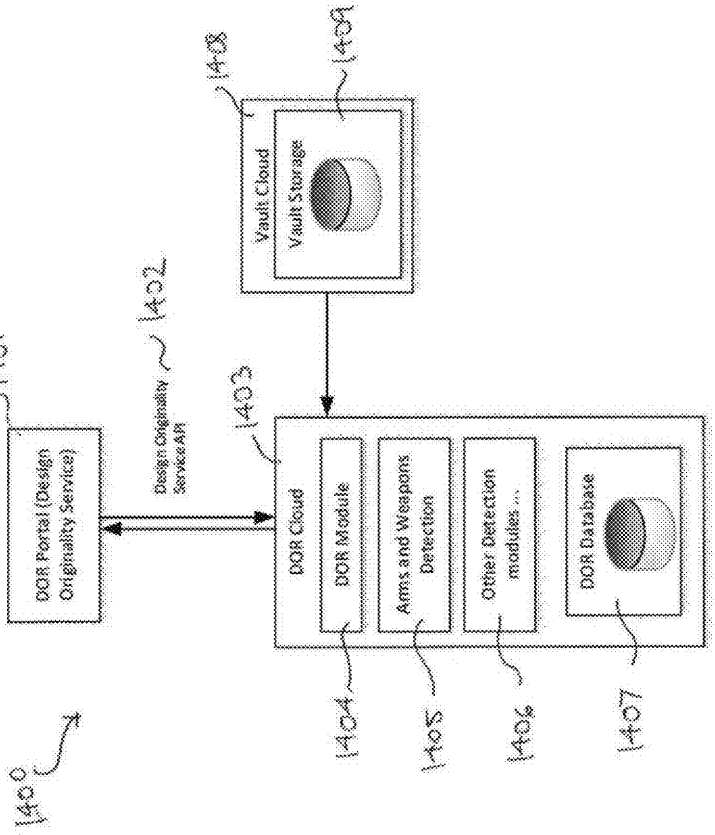

1400
1401
DOR Portal (Design Originality Service)
1402
Design Originality Service API
1403
DOR Cloud
DOR Module
Arms and Weapons Detection
Other Detection modules ...
DOR Database
1404
1405
1406
1407
1408
Vault Cloud
Vault Storage
1409

FIG. 14

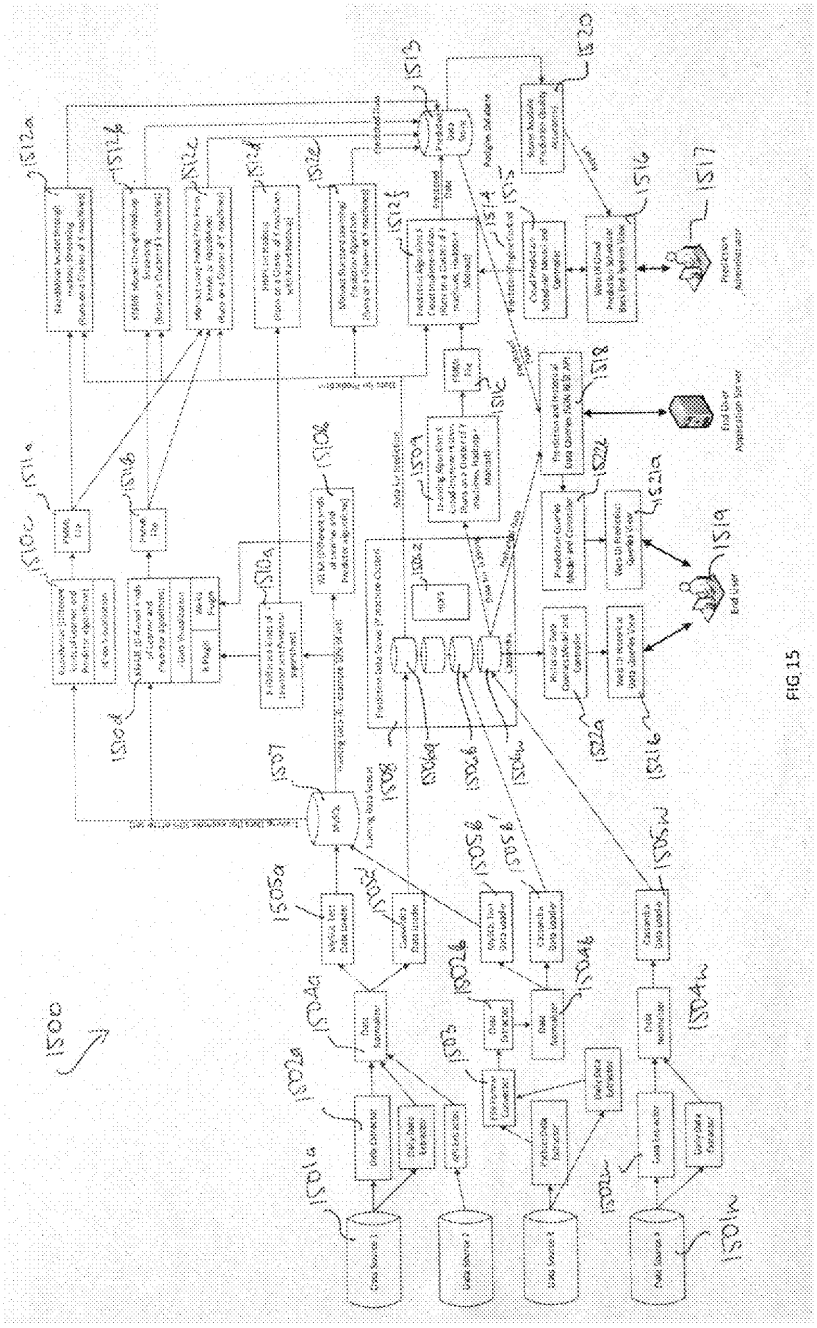


FIG. 15

METHOD AND SYSTEM FOR ENFORCING 3D RESTRICTED RIGHTS IN A RAPID MANUFACTURING AND PROTOTYPING ENVIRONMENT

RELATED APPLICATIONS

[0001] This application claims priority to co-pending EP12181430.5-1243 filed Aug. 22, 2012, the contents of which are hereby incorporated by reference in their entirety.

FIELD OF THE INVENTION

[0002] The principles of the present invention relate to electronic devices, such as rapid manufacturing devices and systems, either by additive or subtractive methods, including 3D printing devices, with intellectual property rights enforcement features using CAD physical 3D model comparisons to enforce other type of use restrictions.

BACKGROUND

[0003] Rapid manufacturing and rapid prototyping are relatively new class of technologies that can automatically construct physical 3D models from Computer-Aided Design (CAD) data. Usually these methods make use of additive manufacturing technologies, including 3D printers.

[0004] 3D printing or additive manufacturing (AM) is a process of joining materials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies, such as traditional machining. Several technologies are available for industrial uses, including for rapid prototyping and rapid manufacturing, but increasingly also for domestic and hobbyist uses. 3D printing is rapidly becoming as widespread as traditional 2D printing had become long ago.

[0005] Combining 3D printing with 3D scanning also makes possible 3D copying, i.e., a process where first a 3D digital model of an object is made by 3D scanning of the object and then a 3D copy of the 3D object is made by 3D reproducing similarly to the process of digital 2D copying.

[0006] It is well known that 2D printing and copying can be used to make copies of copyrighted materials or other materials protected by other intellectual property rights. While some technologies exist to inhibit copying, e.g., documents with security features, such as watermarks, holograms, straps, UV or IR glowing, etc; however, no universal technology exists to control reproducing and copying of copyrighted materials or other protected materials.

[0007] The same challenges are maybe even more important in 3D printing and copying. For example, 3D objects can be protected by number of intellectual property rights, including copyright (e.g., as sculptures, figurines, architectural objects, etc.), industrial design (known in the U.S. as a design patent; e.g., a new shape of a product such as vase or chair), 3D trademark or even by a patent (design patent in the US) or a utility patent. While certain fair use provisions may exist in copyright law (or analogous provisions for design patent or utility patent) allowing in some cases making copies for non-commercial private use, making copies of such 3D objects protected by intellectual property rights is prohibited without a prior explicit permission (a license) from the rights holder.

[0008] Also, no universal technologies exist allowing rights holders to make their IPR protected materials available for 3D reproducing against payment of fee.

[0009] What is needed, therefore, is a method for managing and controlling, including enforcing intellectual property rights and other restrictions in a rapid manufacturing and prototyping environment. Also needed is a 3D reproducing system with intellectual property rights management feature.

SUMMARY

[0010] One aspect of the invention is a method for enforcing 3D intellectual property rights in a rapid manufacturing and prototyping environment for manufacturing 3D objects. The method may include receiving, by a computer system, an inquiry whether a mechanical reproduction of a 3D object or of at least one part of said 3D object is restricted by law or by third person's rights. The 3D object may be presented as an electronic file accessible in a computer system. A determination by the computer system from a 3D restricted rights database as to whether any restrictions exist for the mechanical reproduction of the 3D object or at least one part of the 3D object may be made. A response may be made by a computer to the inquiry, where the response includes information on the restrictions or information that no restrictions exist in 3D restricted rights database. The determination may include performing a 3D object comparison between at least a portion of 3D objects.

[0011] The 3D restricted rights database may include a database of 3D intellectual property rights. The restricted rights database may also include a database of 3D items, the mechanical reproduction of which is prohibited or restricted by law, for example, weapons, firearms or their parts, explosives, etc., or restricted by other types of rights, e.g., privacy rights such as personality rights, rights to personal image (likeness) or contractual rights (e.g., terms of use of a 3D model database).

[0012] According to one embodiment, the inquiry is a request from an end user to reproduce a 3D object by using rapid manufacturing or prototyping tools connected directly or over a computer network to the computer system from an electronic file of the 3D object. The method may include the computer system determining if the 3D object or at least one of its parts is included in the 3D restricted rights database. The determination can be accomplished by comparing the 3D object, the electronic file of the 3D object or a unique identification code of the 3D object stored directly in the 3D object or attached to the electronic file of the 3D object, with corresponding records of the 3D restricted rights database. The method may further include the computer system retrieving such restriction information from said 3D restriction database, and then taking an affirmative action based on such restriction information. Such affirmative action may be one or more actions, including (i) sending a notice to the user that the 3D object is protected, e.g., by intellectual property rights, sending a warning to the user that reproduction of the 3D object is illegal, prohibited or restricted, (ii) stopping the tools from reproducing the 3D object, (iii) stopping the computer system from copying, alternating, moving, removing or streaming of the 3D object and its electronic file, or (iv) offering the user a license for fee or for free.

[0013] According to one embodiment, the method may include offering the user a license or permission to reproduce the 3D object, receiving an acceptance from the user of the terms of the offer, and delivering the 3D object to a rapid manufacturing or prototyping tool over a secure channel. Such method may include receiving a payment from the user by any known means, including but not limited to using credit

or debit card payments, using pre-payment account, using gift cards or vouchers, making a wire transfer using Internet banking, or using electronic payment services, such as PayPal, etc. The method further comprises providing the user, the electronic file, or the rapid manufacturing or the rapid prototyping tool with a unique identification code necessary for reproducing the 3D object.

[0014] According to one embodiment, the inquiry is a request from said rapid manufacturing or prototyping tool, such as 3D printer. Such method may further include providing said rapid manufacturing or prototyping tool with a unique identifier necessary for reproducing the 3D object. The unique identifier is a device specific identifier, associated in a computer system as authorized to reproduce at least one 3D object, and, at the request to print the at least one 3D object, delivering (e.g., sending or streaming) the 3D object to said rapid manufacturing or prototyping tool over a secured data loss proof channel.

[0015] According to one embodiment, the method may include securely storing a list of unique 3D model specific identifying codes on the rapid manufacturing or prototyping tool, such as 3D printer, for checking the restrictions by the rapid manufacturing or prototyping tool itself without the need for continuous real time connection with the 3D restricted rights database. The method may further include updating the list of unique 3D model specific identifying codes from the 3D restricted rights database when the rapid manufacturing or prototyping tool is connected with the 3D restricted rights database over the computer network. Such unique 3D model specific identification codes may be hashes, obfuscated 3D models, Copyright identifiers CIDs, License identifiers LIDs and Anti-Piracy Identifiers APIDs, or a combination of these, or other calculation methods.

[0016] According to one embodiment, the method may include updating the 3D restricted rights database by a rights holder. Such method comprises uploading a new 3D model, representing 3D object, into the restricted rights database, defining part or parts of the 3D object, the use of which is restricted, and providing the terms for reproducing said 3D object or part or parts of the 3D object.

[0017] According to the embodiments of the invention, the inquiry may be received from different sources of 3D models, including, but not limited to a third-party 3D model shop, a 3D scanning device, a 3D computer-aided design software, a 3D model community, a 3D modeling database, etc.

[0018] According to one embodiment, the method may include receiving an inquiry from a 3D scanning device (3D scanner) at the time of scanning a 3D object, and providing the electronic file generated by the 3D scanning device, or the 3D model with a 3D model specific identification code for determining whether reproducing the 3D object is restricted or prohibited.

[0019] The principles of the present invention may also include a system for enforcing intellectual property rights in a rapid manufacturing and prototyping environment. The system may include a source of 3D objects, such as an online shop for 3D models, an online databank of 3D objects, 3D modeling service or other online services, or simply 3D scanning device. The system may further include a restricted rights database, where the restricted rights database may be accessible over an Internet or local computer network (including, but not limited to that the restricted rights database is stored in a "cloud"), an end-user device, connected to the Internet and to a rapid manufacturing and prototyping device (such as 3D printer), which, in turn, is connected both to the end user device and to the Internet, and a restricted rights management application. The 3D restricted rights database may include a 3D intellectual property rights database and a database of 3D items, which mechanical reproduction is prohibited by law. The restricted rights management application may include a 3D objects similarity check module, a Web Application Programming Interface for receiving inquiries from the source of 3D objects, a web based user interface (Web UI), 3D printers API for communicating with 3D printers checking every 3D object to be printed against objects in IPR database, Licensing Module for allowing the user to obtain a license necessary for mechanically reproducing the 3D object, a Royalty Payment Transaction Module allowing the user to pay any license fees necessary, and 3D Object streamer for streaming of 3D object data through Internet or LAN directly to the rapid manufacturing or prototyping tool over a secure channel. In this application, the "cloud" is defined as one or many computers or hardware or software computer systems, which store, process and distribute data.

[0020] One embodiment of a method of enforcing 3D restricted rights in a rapid manufacturing and prototyping environment may include, in response to receiving a 3D object data representative of a 3D object, performing, by a computing device, at least one function on the 3D object data to determine a parameter set for each respective at least one function. At least one business rule may be applied to each parameter set for each respective at least one function. At least one algorithm may be performed to determine whether at least a portion of the 3D object matches a rights restricted 3D object. In response to determining that at least a portion of the 3D object matches a restricted rights 3D object, an action may be caused to be taken, otherwise, in response to determining that at least a portion of the 3D object does not match a restricted rights 3D object, the 3D object may be enabled to be rapid manufactured or prototyped.

[0021] In addition, a determination as to whether the at least a portion of the 3D object matches a rights restricted 3D object may include determining a probability factor that the at least a portion of the 3D object matches the rights restricted 3D object. The function(s) may include characterizing the 3D object or portion thereof from the 3D object data. The characterization may include calculating a number of vertices of which the 3D object includes. Applying the at least one business rule may include checking the parameter set and making a determination based on the parameter set. Making a determination on the parameter set may include making a determination that the 3D object is a firearm or weapon. Making a determination may include voting, using the 3D object data set, for a particular 3D object device represented by the 3D object data. The 3D object data may be normalized prior to performing the at least one function. The normalization may include scaling the 3D object data to be comparable to other 3D object data representative of the rights restricted 3D object to which the 3D object is to be compared. A database of rights restricted 3D objects may be data mined using parameters generated from normalizing the 3D object data. Causing an action to be taken may include preventing the 3D object to be rapid manufactured or prototyped.

[0022] One embodiment of a system of enforcing 3D restricted rights in a rapid manufacturing and prototyping environment may include a storage unit configured to store restricted right 3D objects. A memory may be configured to store data. A computing device may be in communication

with the storage unit and memory, and be configured to, in response to receiving a 3D object data representative of a 3D object, perform at least one function on the 3D object data to determine a parameter set for each respective at least one function. At least one business rule may be applied to each parameter set for each respective at least one function. At least one algorithm may be performed to determine whether at least a portion of the 3D object matches a rights restricted 3D object. In response to determining that at least a portion of the 3D object matches a restricted rights 3D object, an action may be caused to be taken, otherwise, in response to determining that at least a portion of the 3D object does not match a restricted rights 3D object, the 3D object may be enabled to be rapid manufactured or prototyped.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] Illustrative embodiments of the present invention are described in detail below with reference to the attached drawing figures, which are incorporated by reference herein and wherein:

[0024] FIG. **1** is an illustration of a system for providing restricted rights management for rapid manufacturing according to one embodiment of the invention;

[0025] FIG. **2** is an illustration of a more detailed system of FIG. **1** according to another embodiment of the invention;

[0026] FIG. **3** is a flow diagram of an illustrative process for providing restricted rights management for rapid manufacturing in accordance with the principles of the present invention;

[0027] FIG. **4** is a block diagram of an illustrative process for processing 3D objects to determine whether the 3D objects or a portion thereof are original and/or have restricted rights associated therewith;

[0028] FIG. **5** is a block diagram of an illustrative architecture for determining design originality and/or restricted rights status of a 3D object;

[0029] FIG. **6** is an illustration of an illustrative triangular facet that may be used in determining originality and/or restricted rights of a 3D object;

[0030] FIG. **7** is an illustration of an illustrative 3D object with triangular facets representing the skeleton of the 3D object;

[0031] FIG. **8** is a flow diagram of an illustrative system and process for use in classifying 3D objects;

[0032] FIG. **9** is an illustration of an illustrative object skeleton for use in determining design originality and/or restricted rights status of a 3D object;

[0033] FIG. **10** is an illustration of an illustrative 3D model represented by triangular facets for a surface of the 3D model;

[0034] FIG. **11** is an illustrative progressive detailization of a 3D object for use in describing a 3D model;

[0035] FIG. **12** is a pair of illustrations that show a 3D object, in this case torus structures, with a complex outer skeleton simplified through use of a truncation process that represents the 3D object;

[0036] FIG. **13** is an illustration of an illustrative automobile with a truncation skeleton of the outer portion of the automobile;

[0037] FIG. **14** is an illustrative architectural view of a software solution according to one embodiment of the invention; and

[0038] FIG. **15** is an illustrative technical view of the software solution according to one embodiment of the invention.

DETAILED DESCRIPTION

[0039] A system for enforcing 3D restricted rights (RR) in a rapid manufacturing and prototyping environment according to one embodiment of the invention is shown in FIG. **1**. The system may include a source of 3D objects **10**, such as an online shop for 3D models, an online databank of 3D objects, 3D modeling service or other online services, a 3D computer-aided design (CAD) software, or simply a 3D scanning device. The end-user can access the source of 3D objects **10** over the Internet, or over a local area network (LAN), using the end user device **13**. The system further comprises restricted rights RR management application **11**. The RR management application **11** comprises a 3D objects Similarity Check Module, a Web Application Interface (Web API) for receiving restricted rights related inquiries from the source of 3D objects **10**, a web based user interface (Web UI), 3D printers API for communicating with 3D printers **15** checking every 3D model to be printed against objects in RR database **12**, Licensing Module for providing the user with necessary license for reproducing the 3D object, a Royalty Payment Transaction Module allowing the user to pay any license fees required, a 3D Object streamer for streaming of 3D object data through Internet or LAN directly to the 3D printer **15**, and a core for operating the RR management application. The RR management application also has a rights holder Web user interface so that the rights holder can upload new 3D objects into RR database and modify rights associated with such 3D objects, using rights holder device **14**. 3D object is a physical object to be manufactured, and 3D model is defined as any computer 3D representation of such 3D object, such as file or files(s) in any of the computer aided design (CAD) file format, STL file(s), or additive manufacturing file format. (for example, 0.3ds, .MDX, 0.3CT, ACIS, ArchiCAD library part, BE-Bridge, CAD data exchange, COLLADA, AutoCAD DXF, Design Web Format, DGN, .dwg, Geometric Description Language, IGES, KernelCAD, Open Design Alliance, OpenCTM, Parasolid, PLY, PRC, Product data record, Revizto, STL, VDA 6.1, VDA-FS, Wavefront .obj). It can also be one or more files providing views of the 3D object in any image file format.

[0040] The system as shown in FIG. **1** may be used as follows. The user may request a 3D object using end-user device **13** from a source of 3D objects (step **1**) to be reproduced using the 3D printer **15** (step **2**). The source of 3D objects sends an inquiry to the RR management application through the WebAPI (step **3**). In one embodiment, the 3D printer includes a software code or a hardware device installed that requires, through a 3D printers API, for every 3D model to be printed with a confirmation from the RR management application that reproducing of the 3D model is not restricted (step **4**). The restricted rights management application checks if restrictions exist in RR database (step **5**). If so, the Licensing Module is initiated (step **6**), and the end-user is offered a license. The Licensing Module may be adapted to provide any type of license, including single, multiple or other types of licenses for reproducing authorized restricted rights 3D models. The end-user can communicate with the RR management application via the end-user device **13** through the user Web UI (step **7**). If the license terms include royalty payments, the Royalty Payment Transaction Module is initiated and the end-user is provided with opportunity to pay for the license. Then, the 3D printer **15** is provided with a confirmation that the 3D model can be printed,

and the 3D model is delivered (sent, streamed) to the 3D printer through 3D Object streamer (step **8**).

[0041] The system allows the rights holder via a rights holder device **14** to update the RR database to upload new 3D objects together with their restriction terms (step **9**).

[0042] The WebAPI can support authentication, secured 3D model transfer, or other restricted rights requests. The Web API can be implemented, e.g., using Secured SOAP, REST, HTTP with, e.g., OAuth (open source) authentication, or other suitable means.

[0043] The 3D streamer can be implemented, e.g., using RTMPS (real time messaging protocol over a secure SSL connection using HTTPS), FDT (fast data transfer), 3TP (an application layer protocol for streaming 3D graphics), a proprietary protocol developed for restricted rights management application, or other suitable means.

[0044] In accordance with another embodiment of invention, a more elaborated system of FIG. **1** for enforcing 3D restricted rights in a rapid manufacturing and prototyping environment is shown in FIG. **2**. The same reference numbers are used as in FIG. **1**. The skilled person appreciates that the teaching of the embodiments of FIG. **1** and FIG. **2** can be freely combined with each other. The system comprises a source of 3D objects **10** as in FIG. **1**. The end-user can access the source of 3D objects **10** over the Internet or LAN using the end-user device **13**. The restricted rights RR management application **11** may include a 3D objects Similarity Check Module, a Web Application Interface (Web API; for checking the licenses/copyright/anti-piracy or other restricted rights permissions status and authorizing of 3D print requests from 3D online services and communities) for receiving IPR related inquiries from the source of 3D objects **10**, a web based user interface (User Web UI), 3D printers API (for verifying, transferring and granting licenses to parts of or to complete 3D models) for communicating with 3D printer **15**, checking every 3D model to be printed against objects in RR database **12**, Licensing Module for providing the user with necessary license for reproducing the 3D object, a Royalty Payment Transaction Module allowing the user to pay any license fees required, a 3D Object streamer for streaming of 3D object data through Internet or LAN directly to the 3D printer **15**, a module for calculating, attaching and storing unique identification code (such as copyright identifier CID, license identifier LID, anti-piracy identifier APID, etc), a secured API for updating 3D printer storage with 3D models unique identifiers, Module for calculating geometric hashing algorithms, Module for restricting the unauthorized reproduction of 3D items prohibited by law, and a core for operating the RR management application. The RR management application also has a rights holder Web user interface so that the rights holder can upload new 3D objects into RR database and modify rights and permissions associated with such 3D objects using rights holder device **14**. The RR management application has a restricted parts selection tool UI allowing the rights holder to indicate the parts of the 3D object which use is restricted or prohibited.

[0045] 3D Printer as shown in FIG. **2** may include a secured local offline storage of 3D models and unique identifiers, a module for updating the local 3D printer storage with 3D models unique identifiers and a Module for secure local offline calculation of 3D models unique identifiers.

[0046] The system as shown in FIG. **2** is operated similarly to system on FIG. **1**. The user is requesting a 3D object from a source of 3D objects (step **1**) to be reproduced using 3D printer **15** (step **2**). The source of 3D objects sends an inquiry to the RR management application through the Secure WebAPI (step **3**). The 3D printer **15** has a software code or a hardware device installed that requires for every 3D model to be printed with a confirmation, through a 3D printers API from the RR management application that reproducing of the 3D model is not restricted (step **4**). The restricted rights management application checks if restrictions exist in RR database (step **5**). The Licensing Module is initiated (step **6**), and the end-user is offered a license, or the Module for restricting the unauthorized reproduction of 3D items prohibited by, law is initiated (step **6**A), if the 3D object is prohibited from reproduction. The Licensing Module may be adapted to provide any type of license, including single, multiple or other types of licenses for reproducing authorized restricted rights 3D models. The end-user can communicate with the RR management application through the user Web UI (step **7**). If the license terms include royalty payments, the Royalty Payment Transaction Module is initiated and the end-user is provided with opportunity to pay for the license. Then, the 3D printer **15** is provided with a confirmation that the 3D model can be printed, and the 3D model is delivered (sent, streamed) to the 3D printer **15** through 3D Object streamer (step **8**).

[0047] The 3D printer **15** is adapted for offline storing of 3D models and their unique identifiers in a storage. The storage is updated while the 3D printer is connected to the network (step **12**), and the unique identifiers are calculated in the 3D printer or updated from the network.

[0048] The system allows the rights holder to update the RR database to upload new 3D objects together with their restriction terms (step **9**), including determining which parts of the 3D object are restricted from reproduction using Restricted parts selection tool UI (step **11**).

[0049] A method of enforcing 3D restricted rights in a rapid manufacturing and prototyping environment according to one embodiment of the invention is depicted in FIG. **3**. The method may include receiving by a computer system a request to copy (i.e., to reproduce, to make a mechanical 3D copy) of a 3D object (step **300**) or of at least one part of said 3D object; determining by the computer system whether any restrictions exists for mechanical reproduction of the 3D object by comparing the 3D object with objects in a restricted rights database (step **301**). First, it is checked if any IPR right exists (step **302**). If yes, a determination may be made if a license is available (step **303**). If not, the 3D reproducing is stopped or blocked by the computer system. If yes, the user is offered a license to make a 3D reproduction of the 3D object (step **304**). If the license is accepted (this may include paying for the license, if the license is not for free) (step **305**), the 3D reproducing request is confirmed (step **306**). This may include providing the reproducing device, the 3D object itself or the user with a unique ID enabling making one or more reproductions. If the license is not accepted, the 3D reproducing is stopped or blocked (step **311**). Other affirmative actions can be taken by the system as described below.

[0050] If no IPR's exist, other restrictions are checked (step **307**), such as restrictions to make copies of prohibited items such as weapons and firearms and their parts, or restrictions based on other rights such as personality rights, rights to personal image (likeness), or restrictions based on a contract (e.g., access agreement terms to use commercial 3D image databanks). If yes, it is checked if permission is available (step **308**). If not, the 3D reproducing is stopped or blocked. If yes, the end-user is provided with the terms of such permission

(step **309**). If the terms are accepted by the end-user (this may also include paying fees for making 3D reproductions) (step **310**), the 3D reproducing request is confirmed (step **306**). If the terms are not accepted, the 3D reproducing is stopped or blocked (step **311**).

[0051] The determining if any restrictions exist in a restricted rights database can be accomplished by comparing the 3D object, the electronic file of the 3D object, or an identification code stored directly in the 3D object or attached to the electronic file of the 3D object, with corresponding records in the 3D restricted rights database.

[0052] The computer system may take an appropriate affirmative action based on the restrictions. Such affirmative action may be one or more actions, including (i) sending a notice to the end-user that the 3D object is protected by intellectual property rights, (ii) sending a warning to the user that reproduction of the 3D object is illegal, prohibited, or restricted, (iv) stopping the tools from reproducing the 3D object, (iii) stopping the computer system from copying, alternating, removing, moving or streaming of the 3D object and its electronic file, providing the reproduced 3D object with labeling that it is an unauthorized copy, is for private use only, or other information of similar nature, and (v) altering the reproduced 3D object to make it safe or unusable for prohibited purposes, or offering the user a license for fee or for free.

[0053] Offering the user a license to reproduce the 3D object, receiving the user's acceptance of the offer and delivering (sending, streaming) the 3D object to a rapid manufacturing or prototyping tool over a secure and data loss proof channel. Such method may include receiving a payment from the user by any known means, including but not limited to using credit or debit card payments, using prepayment account, using gift cards or vouchers, making a wire transfer using Internet banking, or using electronic payment services such as PayPal, etc.

[0054] As previously described, one embodiment of the 3D reproduction process may be controlled by providing the end-user the electronic file, the rapid manufacturing, or the rapid prototyping tool with a unique identification code necessary for reproducing the 3D object. Alternative processes may be utilized in accordance with the principles of the present invention as provided further herein.

[0055] Several use cases of the systems and methods according to present invention are now discussed.

[0056] Use case 1: The Rights Holder

[0057] A rights holder of any restricted rights, including intellectual property rights, such as copyright, design right, 3D trademark right, design patent or utility patent rights, personality rights, or other rights, opens an account with the service. The rights holder agrees to its terms. Using the rights service, accounts can be for fee or for free. The rights holder can be asked to fill in payment options and details. There can be several security and authentication alternatives available to make sure that the rights holder has the right to register the designs as well as for future authentication of returning rights holders.

[0058] In one embodiment, rights holder can open a modifiable storefront in the service. This storefront user interface can be branded to reflect the rights holders wishes, brands and requirements and for display to visiting customers of the service.

[0059] The rights holder may upload a 3D model into the service and indicate the parts of the 3D object or the whole 3D object that rights holder wants to protect by limiting its 3D copying. Any licenses can be revoked or canceled any time, due to misuse or other reasons.

[0060] The rights holder may select the desired protection or restrictions level. For example, the rights holder may choose to allow free and unlimited distribution of the 3D model (but, e.g., requiring that attribution rights are respected) or to charge a license fee for parts or the whole 3D object or their combinations.

[0061] The service may calculate and categorize the 3D objects based on their shape, color, pattern and selected other properties that the rights holder wants to protect or their combinations. The identification data is stored into the restricted rights database.

[0062] The service algorithms are used to filter out basic forms, shapes, patterns, etc., and other objects that cannot be protected, such as a simple square or round object. These can be either preloaded into the service by the service administrator or can be updated, e.g., by the rights holders.

[0063] Service algorithms can be used to limit submission of unauthorized claims for already uploaded and protected 3D objects by other service users so that basic forms and objects of other rights holders cannot be claimed by unauthorized service users. In case of conflict, the service can present options to the service users or rights holder to resolve the conflict and alter the option selections until the submission is approved. A dispute resolution option may be provided. Once there is no conflict, the 3D object is approved for upload and storage in the restricted rights database.

[0064] There can be a community feedback tool for the rights holders and other service users to vote, rate and give feedback on the uploaded 3D objects. This feedback can be about the quality of the design, model and their printability or objections to the originality of the design or model that has been uploaded and protected. There can be an elimination process based on enough negative feedback of the protection or restrictions of the whole model or parts of it In the case of crowd sourced community projects or other common creation designs, the 3D models and designs can be attributed to several rights holders and royalties and revenues can be attributed to more than one rights holder.

[0065] The service user interface tool allows sharing the submissions during submission process with other rights holders and service users for input, either live during submission process or through accessing the submission 3D model from their own user accounts later, modify the submission and save it.

[0066] The submitting rights holder (e.g., a 3D model store owner in the service) can change roles and give the submission role to another rights holder or service user at any stage. The service saves and keeps track of these changes. There can be multiple rights holders and other users registered as the authorized submission/administration role.

[0067] Use Case 2: 3D Models Online Shopping Site Customer

[0068] 3D online shopping site customer or user is a person using the service from a third-party or service's own shopping web service, community, or application for 3D objects.

[0069] The visitor of such websites and services can use features of the service as a registered or unregistered user. If the user registers as a user of a third-party service, the process depends on their rules and terms.

[0070] If these third party shopping sites use the system according to the principles of the present invention, their services are connected to it through the network using the APIs.

[0071] If the user accesses the service from a computer (or other device), the service can collect identifying information about the device, such as location and IDs, including the 3D printer ID if one is detected. If there is no 3D printer ID detected, the 3D printer is provided with the ID by the system. If the ID of a 3D printer is not automatically detected and sent to the service, the service can provide a unique ID to the 3D printer for IPR purposes and store the unique ID. If the user computer is connected to a 3D printer or the 3D printer is connected directly to the network, the 3D printer ID is registered into the service databases for future verification purposes through the network. This ID can be used as an additional method for verifying authorization to reproduce objects through that 3D printer.

[0072] 3D Object reproducing licenses can be tied with printer IDs to have the option to restrict further the printability on certain printers.

[0073] When a user is visiting a third-party online service or community for printable objects, the user may be presented with a range of objects to review and purchase. The service may learn from these visits, habits, and purchases and may suggest options based on these as well. The 3D printer ID can be used to help improve these results.

[0074] The user selects one or more objects for 3D reproduction. The user is presented with the payment options and completes payment. This can happen inside the service, in an external service or through online payment brokers. In the case of a user visiting a 3D print shop and requesting to print the object there or sending the object to such a service, these steps may be performed by the administrator of the service.

[0075] Reproduction rights can be verified in different stages:

[0076] A third-party service can allow the service to check their uploaded 3D objects for restrictions during their upload by users of the third-party service and/or the restrictions may be checked every time the 3D object is reproduced. The 3D objects may be verified at least each time they are reproduced by the 3D printer.

[0077] Open Source 3D printers can be restricted, controlled, and verified by attributing IDs to them or checking the 3D printer ID against stored IDs in service database as an effort to control also open source printers when they are connected to the networks, such as the Internet, and are trying to reproduce any 3D object.

[0078] If the user is allowed to buy and download a full 3D model via the network, a user can download the full 3D model to the local computer or device.

[0079] Before reproducing, the 3D printer and/or the computer sends the 3D model to the service where it is compared, in whole or in part, to the 3D models stored in a restricted rights database and restrictions.

[0080] In one embodiment, the service can also divide the 3D models into smaller portions using different methods and send these portions to the service where they are checked against the database. In this case, a full 3D model does not have to be sent to the service for comparison.

[0081] If there is no match found that restricts reproducing in any way, a permission to print it locally is returned from the service and presented to the user. Alternatively, in such case no information is presented to the user and reproducing can proceed automatically.

[0082] If there is a match found in the service and database the user can be offered a license to reproduce it. These licenses, their cost, duration, validity, etc., can vary depending on the rights holders or service provider terms.

[0083] If the third-party or the service uses 3D model streaming over a secure data loss proof channel, the 3D model is checked in the service before it is streamed to the user for reproduction so that unauthorized objects are not streamed. Alternatively, a routing comparison method is used where the streamed data are compared live during streaming in the service against the models in the database by routing the stream via the service before or during the reproduction. When restricted rights are found, the streaming can be stopped or blocked.

[0084] The service can check during the streaming that the process is successful and not modified in any way in breach of the terms of the license.

[0085] Use Case 3: User Own Offline Creation

[0086] In this case, the user is not connected to the Internet, to the service, and/or the restricted rights database. The user creates a 3D model using a computer or other device, software, through modifying an existing model, or by a scanning the 3D object with a 3D scanner.

[0087] The 3D printer can be equipped with software and/or hardware (storage) that locally stores and updates a library/list/collection of objects that are restricted for any reason from reproducing (3D reproducing). These locally stored objects can be but are not necessarily full models. The locally stored objects can be calculations of their originals for restriction verification purposes.

[0088] If the 3D printer is offline from the network, the 3D printer will only allow reproducing objects that are not recognized as restricted in the local storage. If there is a match against the stored models, it is not possible to print the object before reconnecting to a network and updating the storage and possibly purchasing a license for it if there is still a match and availability online.

[0089] A maximum temporal or numerical limit until which the 3D printer can be allowed to be offline and not updated may be set. If this time or printed objects number limit is exceeded, the printer may be configured to not print more objects before updating it again. Such a restricted configuration is to prevent or minimize making unauthorized 3D copies during extended offline use. The time limit or number of printed objects can vary depending on manufacturer, service or other reasons but cannot be changed or tampered with by unauthorized users (consumer, authority or business user).

[0090] Once the 3D printer has network access restored, the local storage is automatically updated. This process can not be modified by unauthorized users. These functions can be combined with the 3D printer ID to further improve accuracy of each printer protection updates. Service databases can keep track of each stored 3D printer and their attributes for IPR protection purposes and store their update history, their offline/online status, etc. The service can produce statistics of global 3D reproducing through these printers. Such result can be used to find out shopping habits, offer recommendations and advertising possibilities. The service can present prompts and reminders to the users when the 3D printer is due for updating as well as warnings about impending limitations due to missing updates.

[0091] Use Case 4: User Own Online Creation

[0092] In this case, the user is connected to the Internet or other computer network and to the service and its databases. The user can create a 3D model using a computer or other device, software, through modifying an existing model, or by scanning the 3D object with a 3D scanner. The 3D printer can check both locally and/or through the network connection to the service if the object or any part of it can be reproduced for free, for a license or cannot be reproduced at all, e.g., when the rights holder has completely restricted the reproducing or the item is a dangerous item, such as a weapon or parts of a weapon.

[0093] The 3D printer can be equipped with software and/or hardware (storage) that locally stores and updates a library of objects that are restricted for various reasons for reproducing. These locally stored objects can be but are not necessarily full models to prevent misuse. In one embodiment, the objects can be only calculations of their originals for copyright verification purposes.

[0094] If the object is not restricted, can be reproduced for free, or cannot be found in the database, the printer will print it without restrictions. There can be a message to be displayed to the user or not.

[0095] If the object is fully or partially licensed, the service will prompt the user to acquire a license to print the object. This license is offered to the user for free or for a fee. The user can be shown different payment options and guided through the payment in easy steps.

[0096] The user can accept or reject the license offer. In the case of accepting the license offer, the user will complete the purchase process. If the user rejects the license offer, the reproducing is not possible for items that require the license.

[0097] When the license is granted (and paid if not free), the service will start delivering (sending or streaming) the 3D model to the 3D printer of the user, and verifies that the 3D model is fully printed. In the case of quality issues, the user can reprint the object, e.g., when the printer runs out of reproducing material or has network problems. The user has a feedback channel to issue a quality claim in case of problems with the service.

[0098] The licenses can be acquired for reproducing single or multiple items. There can be a time limit within which the object must be printed or all the licenses be used. Licenses can be also tied to a particular printer ID and restricted per printer ID in which case the object can only be printed in that particular printer. For other printers, a new license is required—this can depend on the license selection the rights holder chooses or can be part of the service by default.

[0099] 3D model streaming to the 3D printer happens using either standard or proprietary protocols, formats and methods. Interrupted (for example, network connection problems) can be avoided by buffering the model stream to the printer or the computer.

[0100] The service completely prohibits the print or manufacture of parts of fully restricted items, such as guns. These restricted items can be uploaded into the service by authorities, for example. There can be a message displayed to the user that certain items are not allowed to print with or without a license or permission.

[0101] These functions can be combined to the 3D printer ID to further improve accuracy of each printer protection updates. The service databases may keep track of each stored 3D printer and their attributes for IPR purposes and stores their respective update history, their respective offline/online status, etc.

[0102] Use Case 5: 3D Scanner Case

[0103] The user has a 3D scanner, access to a 3D scanner, or orders an object to be 3D scanned in a service. The user selects a 3D object for scanning and scans the object using the 3D scanner. The 3D scanner is connected to the Internet and attaches a tag or tags to the scanned 3D object file with or without a software on the computer or scanner. In offline use, this tagging can happen via locally installed software and/or hardware device on the 3D scanner and/or computer. The user can attribute certain tags to the scanned objects such as shape, color or pattern descriptors or choose to have the service to do this automatically.

[0104] Certain tags can already identify the object as registered, or IPR protected, for example, if scanning a known protected object, or categorize it based on its features, shapes, patterns for recognition then or later in the service database. These tags can include such as shape ("round", "square", etc.), taxonomy ("mouse", "bird", etc.), color ("red", etc.) or any other type of tags redeemed necessary or their combinations. These tags can be hashes or encrypted etc. and not visible to the user.

[0105] Certain tags that are for identifying IPRs are not user modifiable and accessible, they are created automatically and are tamper proof. Tag modification can be allowed for certain administrators. Scanners with IDs can be identified when connected to the network or a computer/network and store the IDs for IPR protection and restriction enforcement purposes.

[0106] With regard to FIG. **4**, a block diagram of an illustrative process **400** for processing 3D objects to determine whether the 3D objects or a portion thereof are original and/or have restricted rights associated therewith is shown. This process could be also called a Design Originality Recognition (DOR). During this process computer system is able to analyze, store and compare 3D objects for copyright recognition and licensing purposes in 3D printing and manufacturing. The illustrative process **400** is shown to include a 3D object **402** to be presented to the process **400** for indicators to be applied thereto. 3D object **402** may be presented as a stereolithographic (STL) file or other format 3D file, as understood in the art. The STL file may include a list of facets of a 3D object to be 3D printed or otherwise manufactured. Each facet may be uniquely identified by a unit normal (i.e., a line perpendicular to a triangle that at least in part defines a facet and has a length of 1.0) and three vertices (i.e., corners of the triangle). The 3D object **402** may be submitted to a 3D object indicators module **404** inclusive of one or more indicators **406***a*-**406***n* (collectively **406**). The indicators **406** are functions that receive an input of the 3D object **402** in any 3D file format (e.g., STL, OBJ, etc.). More specifically, the indicators **406** may be any kind of function that characterizes the 3D object **402** based on its representation in the 3D file.

[0107] Indicators results **408** for the 3D object **402** include results **410***a*-**410***n* (collectively **410**) from respective indicators. Each of the indicators **406** outputs one or more parameters **412***a*-**412***n* (collectively **412**) that characterize a given 3D object from a certain perspective. Note that each of the results **410** of the respective indicators **406** have parameters (e.g., parameters **412***a*'-**412***n*' (collectively **412**'), parameters **412***a*"-**412***n*" (collectively **412**")) that may be the same or different from those of other indicators. If the indicators **406** are all different from one another, then it would be expected

that some or all of the parameters **412**, **412'**, and **412"** resulting from the different indicators **406** would be different by virtue of the functions producing different parameters. Indicators can return similar or the same results regardless of 3D model or 3D object shape and representation (plain object, molds for the object, parts of the object, etc.) For example, it is possible to have 3D models of a gun, and print a gun, however it is also possible to have a 3D model of a mold, which could be used to create the same gun. It is also possible to have separate 3D models, which in case of printed, could be put together to build a gun, or there could be different parts of mold or molds, which after printing and combining could be used to produce the same gun. In all these cases, indicators will give very similar results, so that based on the resulting parameters it is possible to distinguish visually or by the usage of data mining and machine learning algorithms that all 3D object representations actually represent one and the same or very similar object, in this particular example a gun.

[0108] One example of an indicator is a function that calculates a number of vertices of which a 3D object includes. In such an example, a single parameter for the number of vertices exists, where a result for that parameter is a single value (e.g., 3478 vertices). Other more complex functions for the indicators **406** may be used, including center of mass, corpus indicator, detail indicator, comparison of angles, nearest points, number of points, number of edges, number of facets, different types of skeletons, object geometric representation, average deviation of points, bounding corners density, and corresponding parameters all these possible functions return, for example: graphs, vectors, objects, JavaScript Object Notation (JSON), and so forth. These indicators **406** may help the process **400** solve a problem of comparing a subcomponent or whole of a 3D object **402** that is being processed prior to printing or manufacturing in accordance with the principles of the present invention.

[0109] Continuing with FIG. **4**, a 3D objects business rules module **414** may include one or more business rules **416***a*-**416***n* (collectively **416**). The business rules **416** are rules that check the parameters **412** and make a decision based on the parameters **412**. Business rule may be any rule which describes which states one or set of returned indicator functions parameters should take in order to distinguish one model from another or one class of objects from another. Examples of business rules are: check for tubes in the model, based on indicators, if there is a subshape like tube, with hole diameter between 4 mm and 50 mm, then set has_gun_tube parameter to 1; check for rifling inside the tube, if it has it set has_gun_rifling_tube parameter to 1. The parameters may have values of 0 and 1, or may also include a probability coefficient, for example 0.68. Business rules could be simple if-then-else clauses, or more complicated data mining and machine learning algorithms which, based on the indicators values, detect whether a particular object is e.g., a firearm or weapon, detect the type of the object, or, e.g., of an animal, etc.

[0110] The decisions made by the business rules **416** may be generated in a coded way so that it is possible to process the decisions using a classificator module **418** with data mining algorithms **420***a*-**420***n* (collectively **420**). The classificator module **418** is illustrative for use in identifying a portion of or a complete firearm or weapon, and the data mining algorithms **420** support the classificator module **418** in that regard. It should be understood that the classificator module **418** may be configured to identify any other object, including objects that are protected by intellectual property rights (e.g., copyright) or otherwise.

[0111] An example list of business rules may include (i) loading techniques, (ii) magazine types (e.g., internal, detachable, belt-fed, etc.), (iii) firing mechanisms (matchlock, wheel-lock, flint-lock, percussion cap, etc.). Other business rules that may be used to check the parameters **412** for other weapons (or non-weapons) may additionally and/or alternatively be utilized in accordance with the principles of the present invention. The business rules **416** may be configured as simple if-then-else clauses or more complicated data mining and machine learning algorithms which, based on the values of the indicators, detect whether a particular object is a firearm, weapon, or neither, and, if so, which type is it. Makes and models of firearms and weapons may additionally be determined if sufficient parameters are available to the business rules **416**. One of the implications of this method is voting data mining and machine learning algorithms, as understood in the art, for firearms and weapons detection from 3D files. In voting, one or more portions of a 3D model may be voted on by a computing unit, which may be formed of one or more computing devices locally or remotely positioned and in communication with one another, as to whether the one or more portions of the 3D model match a rights restricted 3D model. In determining whether a match exists, matching may be performed using logical and/or mathematical algorithms as described herein.

[0112] As an example of an algorithm, if 3D object has more than 1 million vertexes, then a determination may be made that the 3D model is, for example, a model of animal or a human. Of course, the decision may be performed based on results of many indicators and associated business rules using machine learning and data-mining classification and prediction algorithms. For the algorithms **420** that are configured to identify a firearm or weapon, the algorithms may determine whether the 3D object **402** is representative of a component or an entire firearm (e.g., trigger) or weapon (e.g., detonator) by comparing a number of vertexes, number of facets, angles of the facets relative to one another, and so forth that define the 3D object.

[0113] The classificator module **418** may include known algorithms that are publicly available or proprietary algorithms. Such algorithms may utilize linear regression, time series, Naïve Bayes (learner and predictor), fuzzy rules (learner and predictor), multi-layer perceptron (MLP) neural networks, probabilistic neural networks (PNN), such as k-nearest neighbor (k-NN), decision trees (learner and predictor), boosting (learner and predictor), association rule (learner and predictor), support vector machines (SVM), and so forth.

[0114] As further shown, a database **422** may include data that is supportive information on predicting whether the 3D object **402** is a firearm or weapon. Data may be supplied to the database **422** from the indicators results **408** and supplied to and accessed by the classificator module **418**. The database **422** may utilize any data repository configuration and protocol, such as a relational database, as understood in the art. In one embodiment, the database **422** stores sets of parameters that are representative of results from 3D object indicators, such as 3D object indicators **404**. The sets of parameters may be used for comparison purposes by the algorithms **420** to determine or approximate what, if any, component or whole firearm or weapon is being represented by the 3D object **402**.

While shown with a database **422**, it should be understood that the principles of the present invention may be independent of a database. Final results **424** may be produced from the algorithms **420** to identify the 3D object or its parts to be identified as a firearm, weapon, or component thereof. Again, final results **426** identifying 3D objects other than firearms or weapons, such as 3D objects that are being protected by intellectual property rights, may be identified using different indicators **406** that produce different parameters, different business rules **416**, and/or different algorithms **420**.

[0115] In the event that the restricted rights process **400** is operating within a 3D printer or other manufacturing equipment, in the event that the final results **424** determine that a 3D object has rights that are being protected, as previously described, the 3D printer or other manufacturing equipment may be prevented from printing (or caused to be prevented from printing from a remote computing device, such as an IPR database). In addition, a notification may be made to a manager of the protected rights and/or governmental authorities along with location (if GPS or network address enabled) and other user information if available via a network communication (e.g., email, SMS message, or other electronic message).

[0116] With regard to FIG. **5**, a block diagram of an illustrative architecture **500** for determining design originality and/or restricted rights status of a 3D object is shown. The architecture **500** may include a master database **502** that may receive 3D models **504**. Alternatively, the 3D models **504** may be submitted directly to slave databases **506***a*-**506***n* (collectively **506**) via a communications network (not shown), such as the Internet. As understood in the art, the slave databases **506** may be positioned in the "cloud," thereby enabling the master database **502** to submit job requests to the slave databases **506** utilizing a cloud computing protocol, as understood in the art. The master database **502** and slave databases **506** may be integrated or in communication with a computing system, such as a computer server, that may be configured to perform both data processing (e.g., performing 3D indicator processing) and database management operations.

[0117] A subprocess **507** that may be performed by the slave databases **506** (or computing systems associated therewith) to perform normalization on the 3D models **504**. The subprocess **507**, in this embodiment, is used to generalize or otherwise produce a less detailed representation of a 3D object. The normalization, as provided further below, is capable of adjusting the 3D model (or altering a copy of the 3D model to preserve the integrity of the original model) so that size, orientation, and other parameters may be similarly sized and oriented with other 3D models and/or objects, for example, 3D object could be moved to the beginning of coordinates (x=0, y=0, z=0). Normalization also for example can include resolving errors in CAD file and treating surface errors. As shown, a normalization input **508**, which is a 3D object in an STL or other datafile configured to store a 3D object, may be submitted to a function **510** to compute center of gravity and/or rotate the 3D object. The function **510** may compute the center of gravity of a 3D object to produce a parameter that may thereafter be compared with other 3D objects. In one embodiment, the 3D object may be rotated (or a value indicative of the rotation to align the 3D object to a normal position may be computed). By rotating or calculating a rotation value as a parameter, a more direct comparison can be made between two 3D objects.

[0118] A function **512** may be used to resize the 3D object (i.e., the normalization input **508**). Because datafiles used to store 3D objects may have different scales, resizing the 3D object to be "one size" may better enable two different 3D objects to be compared. In one embodiment, standard dimensions in which the 3D object may be fit may be utilized. In one embodiment, the standard dimensions may be different depending on a variety of factors, such as scale, complexity, type of 3D object, and so on. As an example, vehicles may be sized to standard dimensions in which automobiles typically are scaled, while semiconductor devices may be scaled for standard dimensions in which semiconductor devices are typically scaled.

[0119] Function **514** may be configured to generate a corpus indicator, where the corpus indicator may be considered a box that is used to fit the 3D object. The corpus indicator may use the standard dimensions used by function **512**. It should be understood that having the standard dimensions identical for each 3D object, or at least each similar type of 3D object, may be helpful for comparison purposes, but that differences between size and scaling may be possible for performing comparisons between 3D objects in accordance with the principles of the present invention.

[0120] Function **516** may be a detail indicator configured to set normals of 3D objects to be aligned. For example, a 3D object that is an automobile may have a normal extending from a roof of the automobile, and 3D objects of automobiles may be processed by the detail indicator to cause the normals to extend from a roof of the automobiles and oriented in the same direction (e.g., pointing up at zero degrees). The detail indicator may be a parameter representative of the orientation adjustment needed to have a normal of the 3D object positioned aligned with other normals of 3D objects. By orienting the normals of 3D objects, easier comparisons can be made between 3D objects.

[0121] Function **518** may be configured to classify one or more parameters of the 3D object or components thereof. The classes may be assigned predetermined classes (e.g., class 1, 2, . . . , n; class A, B, . . . N) to the 3D object. The classes may reflect similarity of parameters between the 3D object or components thereof that may be used for data mining or comparison purposes. A normalization output **520** include parameters that represent the generalized 3D object. The normalization output **520** may be coded in a manner that can provide for data mining and/or searching to determine whether the normalized 3D object matches another 3D object or portion thereof.

[0122] Output parameters **522** may include the parameters generated by the various functions in the subprocess **507** used to normalize 3D objects. The output parameters **522** may be saved to the slave databases **506**. In one embodiment, a 3D models parameter database **524** may be configured to store the output parameters **522** of the 3D models **504**. The database **524** may further be configured to store 3D models that have restricted rights, such as intellectual property rights and/or are restricted due to being firearms, weapons, ammunition, etc. The slave databases **506** are in the cloud and may include several, hundreds, or thousands of computers configured to process the 3D model data. As shown, the comparison process **400** of FIG. **4** allows for comparison of the 3D objects with the restricted rights 3D objects, as previously described. The 3D models **504** and/or output parameters of the 3D models **504** (and/or 3D objects) may be communicated and stored in a 3D models database **526** for further usage.

[0123] With regard to FIG. **6**, an illustration of an illustrative triangular facet **600** that may be used in determining originality and/or restricted rights of a 3D object is shown. The facet **600** may be used to define indicators, where each of the three vertices **602***a*-**602***c* and normal **604** may be defined by three coordinates (e.g., x, y, z coordinates). The indicators, which may be a set of coordinates for the facet, may thereafter be used for determining originality and/or restricted rights of a 3D object by comparing the indicators with indicators of another 3D object. The facet **600** is an STL format facet specification that provides for direction of the normal to the facet. The normal vector is calculated mathematically, and provided as a parameter for the facet along with parameters of the vertexes of the facet that may thereafter be used for comparison purposes. In particular, the orientation of a facet is determined by the direction of the unit normal and the order in which the vertices are listed.

[0124] With regard to FIG. **7**, an illustration of an illustrative 3D object as a solid (cube) **700***a* and as a wireframe 3D object **700***b* represented with triangular facets defining the skeleton of the solid 3D object **700***a* is shown. Simplified object skeleton is a set of vertices where each vertex is connected to at least one other vertex. The skeleton can be represented as a graph that is why skeleton comparison is reduced to comparison of graphs. It is noted that vertices pairs are the same as well as the length of links between them. The first problem can be solved by checking isomorphism of the graphs that is true only when "any two vertices u and v of G are adjacent in G if and only if f(u) and f(v) are adjacent in H". In addition to isomorphism, lengths of the links are to be compared. This comparison can be done by checking whether two same vertices have link of the same size (in case objects have different scale this can be done proportionally). The triangular facets may be created using any facet generator, as understood in the art. Although the application provides for the use of triangles, the principles of the present invention may utilize any geometric shape to fit within surfaces or bodies of a 3D object. Conventional 3D model comparison algorithms are limited to comparing only ASCII STL 3D models. As understood in the art, ASCII STL 3D models are limited to a lot of triangles. These triangles form a model, for example, such as a cube, as provided in the wireframe 3D object **700***b*. The principles of the present invention, however, may use an algorithm that compares vertices or vertexes of triangles of one model with the vertices of triangles with another model. As triangles could be in different orders for the same model, the principles of the present invention use the vertices of sub-figures (e.g., triangles) and represent them as vertices of a 3D model. Such technique provides the ability to compare models by their vertices on the plane and pack them. Instead of comparing all vertices for cube, the vertexes are packed to main points. All these points are seen in the solid 3D object **700***a*.

[0125] In one embodiment, the process may read input STL files from first and second 3D models. The models may be into triangles or facets that define wireframes of the surfaces of the 3D models. Triangles may be compared for interconnected vertices recognition. Adjacent vertexes may be grouped into a list of points (e.g., x, y, z). The above steps may be repeated for another 3D model. The whole list of vertices of the first 3D model may be compared with the list of vertices of the second 3D model. Similarity of the two 3D models using the lists of vertices may be mathematically described as a percentage to show similarity of the two 3D models. For example, the a similarity value of 87% may mathematically describe similarity between two 3D models.

[0126] With regard to FIG. **8**, a flow diagram of an illustrative system and corresponding process **800** for use in classifying 3D models is shown. In general, in order to compare two 3D models, descriptors may be used to classify each of the 3D models. The system may transform any kind of 3D model (i.e., described in any 3D file format) using common descriptors as other 3D models dynamically (i.e., on the fly) and compare the descriptors of a 3D model to be compared with existing descriptors of 3D model objects. Such complex processes may use "cloud" technologies, as understood in the art.

[0127] As shown, a 3D model **802** to be 3D printed is shown. The 3D model may be transformed using a transformation process **804** to produce an object descriptor **806** of the 3D model. The transformation process **804** may generate triangles that define surfaces, for example, to create a wireframe of the 3D model and 3D objects thereof. The object descriptor **806** is invariant to transformations, such as scale, rotation, mirror, and translation. A 3D models database **808** may be configured to store known and/or protected 3D models. The same or similar transform **810** as the transformation process **804** may be utilized to transform the 3D models being stored by the 3D models database **808** to produce 3D object descriptors for storage in a 3D object descriptors database **812**. By using the same or similar transform **810** as the transformation process **804**, an "apples-to-apples" comparison may be performed. It should be understood that transformation processes that are different from one another for speed or other purposes may be utilized in accordance with the principles of the present invention. For example, the transformation process **804** may be a faster transformation process and/or have less resolution than the transform **810** as the transformation process **804** may be performed in real-time and have to be consumer acceptable. If a determination is made that a probabilistic match is made between the 3D model and a 3D model stored in the 3D models database **808**, then a more precise transformation may be utilized for the transformation process **804**. The probabilistic match may, for example, be 80% or higher that the 3D object matches a 3D object in the 3D models database **808**.

[0128] A search **814** may be performed via a cloud comparison or other application **816** to compare at **818** the object descriptor **806** with the 3D object descriptors generated from the 3D models being stored in the 3D models database **808**. As understood in the art, the search may be utilized in data mining. The search may be responsive to a search query or be performed on a periodic or aperiodic basis to collect 3D model data based on parameters generated by the processes of FIGS. **4** and **5**. As provided above, the comparison of the 3D model descriptors may result in a percentage of similarity the 3D model **802** or portion thereof and 3D model(s) in the 3D model database.

[0129] In one embodiment, an identifier, such as bracket, associated with the 3D model **802** that provides a basic type of object that the 3D model represents, so that the cloud comparison application can be limited to brackets as opposed to other devices, such as cups, housings, or otherwise. The identifier may for additional detail, such as bracket: server bracket: 4-post server bracket, thereby further narrowing the scope of the search. In one embodiment, a specified hierarchical listing of identifiers may be made available to allow for 3D models to be specified.

[0130] With regard to FIG. **9**, an illustration of an illustrative object skeleton **900** for use in determining design originality and/or restricted rights status of a 3D object is shown. In a 2D format, the object skeleton **900** shows to be relatively simplistic, and real calculation happens with spheres in 3 dimensions. In this embodiment, the object skeleton **900** is configured as a focus of centers of maximal 3D balls **903***a*-**903***n* (collectively **903**) contained within a 3D object. That is, the centers **902***a*-**902***n* of the maximal 3D balls **903** or volume pixels ("voxels") are positioned at local centers of a path of a 3D object. A series of voxels form the object skeleton **900**. That is, the object skeleton **900** may be formed out of smallest units of measurement. Alternative mathematical techniques for determining voxel position, such as using geometric mean along a path, may be utilized.

[0131] With regard to FIG. **10**, an illustration of an illustrative 3D model **1000** represented by triangular facets **1002** for a surface of the 3D model **1000** is shown. Within the 3D model is a skeleton **1004** defined by voxels. This skeleton **1004** shows how even complex models can simplified by the use of a skeleton.

[0132] With regard to FIG. **11**, an illustrative progressive detailization of a 3D object **1100**, in this case a cow, for use in describing a 3D model is shown. A first skeleton **1102** may have little detail of the 3D object **1100**. A second skeleton **1104** may have additional detail of the 3D object **1100**. A third skeleton **1106** may provide more detail of the 3D object **1100**. A fourth skeleton **1108** provides yet more detail of the 3D object **1100**. A fifth skeleton **1110** adds even more detail of the 3D object **1100**. These levels of detail may be generated through use of one or more 3D object indicators that produce results with different deal of parameters. The different levels of detail for each of the different skeletons **1102**-**1110** allows for business rules and algorithms (see, FIG. **4**) to determine that the 3D model is an animal and, as the model detail increases to skeleton **1110**, is a cow. As a result of the progressive detailization,

[0133] With regard to FIG. **12**, an illustration of a pair of torus structures **1200***a* and **1200***b* with a complex outer skeleton simplified through use of a truncation process that represents the 3D object is shown. One reason for using an outer skeleton is that inner skeletons may not always be possible or desirable when a 3D object is scanned using a 3D scanner, for example. The truncation process may utilize a modified process of FIG. **4**, but rather than using a skeleton, such as that shown in FIG. **11**, uses functions that produce layers with certain geometry, such as triangular facets that form surfaces. Torus structure **1200***a* is shown with 5000 modeling elements, such as triangular facets, while torus structure **1200***b* is shown with 500 modeling elements. The resolution difference between the two different torus structures **1200***a* and **1200***b* is relevant as a higher probability of determining a copy of a protected rights 3D object is higher with higher resolution.

[0134] With regard to FIG. **13**, an illustration of an illustrative automobile **1300** with a truncation skeleton **1302** of the outer portion of the automobile is shown. In some cases, the use of the truncated skeleton **1302** may be sufficient to (i) determine that the 3D object is a vehicle and (ii) determine the type of vehicle. Still yet, if a component of the vehicle is to be 3D printed or otherwise manufactured, then a determination may be made as to whether rights restrictions exist for the component. Again, in order to determine such detail, sufficient parameters have to be available as a result of the indicators or functions used to create the resulting parameters, as provided in FIG. **4**.

[0135] A cloud computing platform could be used for realization of processes described above. Cloud technologies for example Hadoop, Cassandra, etc. could be used for this platform, however, non-cloud software frameworks and programming languages also could be used. Depending on business needs, a different final setup of the computer system which implements the process described above could be done. For example, it could be configured only for distinguishing firearms and weapons based on 3D model of the object. It could be setup in a way that all classes of objects are defined, for example, architecture, vehicles, aircrafts, furniture, plants, food, animals, electronics, characters, weapons, anatomy, accessories, DNA and molecules, etc.

[0136] The architectural view of the software solution according to one embodiment **1400** is provided on FIG. **14**. There is a Design Originality Recognition (DOR) Portal **1401** for providing design originality service. This is an end point for customers to access the service through, e.g., a web interface. There is an API **1402**, which is an end point to DOR cloud solution **1403**. DOR Cloud has several modules. DOR module **1404** which includes methods described above. Some of the methods may be implemented separately, for example Arms and Weapons detection module **1405** could be implemented separately in order to load balance the cloud, as for example in real application arms and weapons detection will have more specific indicators and more specific business rules that are not needed to be applied to other 3D models. Other detection modules **1406** could be implemented in a similar way. There is a DOR database **1407**, which stores intermediate and final results of DOR processes. 3D model could be loaded from DOR portal **1401** or from any other storage including Vault Cloud **1408** and Vault Storage **1409**. Vault Cloud stores 3D models in a secured or unsecured way, every 3D model could be split into many pieces, and each peace may be encrypted with its own encryption key. DOR cloud can consist of one or many (e.g., hundreds or thousands) hardware or virtual computing units.

[0137] The technical view of the software solution for 3D models comparison according to one embodiment **1500** is provided on FIG. **15**. Technologies and types of software are not limited or fixed, FIG. **15** shows just one example of implementation. Data sources **1501***a*-**1501***n* are any sources of 3D models, for example CAD files, STL files, etc. from different locations, for example Data Source 2 could be Thingiverse database or web site, Data Source 1 could be a database with 3D models, Data Source 4 could be a computer file storage. There is a set of extractors **1502***a* to **1502***n* and file format converters **1503** when needed. 3D models that are shown as Data on the FIG. **15** are extracted and normalized after conversion. Normalization modules **1504***a* to **1504***n* shown on FIG. **15** also include indicators calculation software. Using Database loaders **1505***a* to **1505***n* 3D models and results of indicators are loaded to corresponding databases **1506***a* to **1506***n*, it could be a distributed database like Cassandra, just single database, storage or cloud storage, like HDFS **1506***z*. There are several types of databases (test database and prediction database), where 3D models and resulting parameters of indicators are loaded. Training Data Subset **1507** shown as MySQL on the FIG. **15** is used for storing training data for data mining and machine learning algorithms training modules implemented in different types of

frameworks. The rest of the data is stored in Prediction Data Server database **1508**. This database is also used to provide training data for proprietary training and prediction algorithm (Learning Algorithm X cloud implementation) **1509**.

[0138] Data usually goes to different open source and non-open source implementations of data mining and machine learning algorithms **1510***a*, **1510***b*, **1510***c* and **1501***d* (for example R, KNIME, RapidMiner, Weka, etc.), which training modules or setups prepare PMML files **1511***a*, **1511***b* and **1511***c*. PMML files define models of predictive analytics and data mining. PMML files are used to transfer training information to prediction modules or setups of data mining and machine learning algorithms **1512***a* to **1512***f* (for example KNIME, RapidMiner, Mahout, custom Hadoop algorithm, RHIPE, or proprietary, etc). Predicted data is stored to Predicted Data Store **1513**, which consists of information about 3D model classes, types, differences between them, similarities, originalities, possible arms and weapons, etc. Prediction engine **1514** is controlled through Cloud Prediction Scheduler Model and Controller **1515**, and is accessible by a user, who is predictive administrator **1517** through Web UI Cloud Prediction Scheduler Back End System View **1516**. There could be an API, Prediction and Historical Data Queries JSON REST API **1518** for getting predicted information from other applications, servers, services, etc. Using Web API **1521***a* and **1521***b* end user **1519** can make requests both to 3D models historical datasets using Historical Data Queries Model and Controller **1522***a*, and to prediction databases using Prediction queries model and controller **1522***b*, so that end user can assess final results, and if needed, tune parameters for algorithms, etc in a prediction administrator role. There is a Scorer Module **1520**, which make prediction quality assurance and reports to Prediction Administrator **1517**.

[0139] The previous detailed description is of a small number of embodiments for implementing the invention and is not intended to be limiting in scope. One of skill in this art will immediately envisage the methods and variations used to implement this invention in other areas than those described in detail. The following claims set forth a number of the embodiments of the invention disclosed with greater particularity.

What is claimed:

1. A method of enforcing 3D restricted rights in a rapid manufacturing and prototyping environment, said method comprising:

in response to receiving a 3D object data representative of a 3D object, performing, by a computing device, at least one function on the 3D object data to determine a parameter set for each respective at least one function;

applying, by the computing device, at least one business rule to each parameter set for each respective at least one function;

performing, by the computing device, at least one algorithm to determine whether at least a portion of the 3D object matches a rights restricted 3D object; and

in response to determining that at least a portion of the 3D object matches a restricted rights 3D object, causing, by the computing device, an action to be taken, otherwise, in response to determining that at least a portion of the 3D object does not match a restricted rights 3D object, enabling, by the computing device, the 3D object to be rapid manufactured or prototyped.

2. The method according to claim **1**, wherein determining whether the at least a portion of the 3D object matches a rights restricted 3D object includes determining a probability factor that the at least a portion of the 3D object matches the rights restricted 3D object.

3. The method according to claim **1**, wherein performing at least one function includes calculating a number of vertices of which the 3D object includes.

4. The method according to claim **1**, wherein applying at least one business rule includes checking the parameter set and making a determination based on the parameter set, and wherein making a determination on the parameter set includes making a determination that the 3D object is a firearm or weapon.

5. The method according to claim **4**, wherein making a determination includes voting using the 3D object data set for a particular 3D object device represented by the 3D object data.

6. The method according to claim **1**, further comprising normalizing the 3D object data prior to performing the at least one function.

7. The method according to claim **6**, wherein normalizing includes scaling the 3D object data to be comparable to other 3D object data representative of the rights restricted 3D object to which the 3D object is to be compared.

8. The method according to claim **1**, wherein causing an action to be taken includes preventing the 3D object to be rapid manufactured or prototyped.

9. A system of enforcing 3D restricted rights in a rapid manufacturing and prototyping environment, said system comprising:

a storage unit configured to store restricted right 3D objects;

a memory configured to store data;

a computing device in communication with said storage unit and memory, and configured to:

in response to receiving a 3D object data representative of a 3D object, perform at least one function on the 3D object data to determine a parameter set for each respective at least one function;

apply at least one business rule to each parameter set for each respective at least one function;

perform at least one algorithm to determine whether at least a portion of the 3D object matches a rights restricted 3D object; and

in response to determining that at least a portion of the 3D object matches a restricted rights 3D object, cause an action to be taken, otherwise, in response to determining that at least a portion of the 3D object does not match a restricted rights 3D object, enable the 3D object to be rapid manufactured or prototyped.

10. The system according to claim **9**, wherein said computing device, in determining whether the at least a portion of the 3D object matches a rights restricted 3D object, is further configured to determine a probability factor that the at least a portion of the 3D object matches the rights restricted 3D object.

11. The system according to claim **9**, wherein said computing device, in performing at least one function, is further configured to calculate a number of vertices of which the 3D object includes.

12. The system according to claim **9**, wherein said processing unit, in applying at least one business rule, is further configured to check the parameter set and make a determination based on the parameter set, and wherein making a deter-

mination on the parameter set includes making a determination that the 3D object is a firearm or weapon.

13. The system according to claim **12**, wherein said processing unit, in making a determination, is further configured to vote, using the 3D object data set, for a particular 3D object device represented by the 3D object data.

14. The system according to claim **9**, wherein said processing unit is further configured to normalize the 3D object data prior to performing the at least one function.

15. The system according to claim **14**, wherein said processing unit, in normalizing, is further configured to scale the 3D object data to be comparable to other 3D object data representative of the rights restricted 3D object to which the 3D object is to be compared.

* * * * *

Appendix 6

VI

K. Isbjörnssund and A. Vedeshin. Secure streaming method in a numerically controlled manufacturing system, and a secure numerically controlled manufacturing system, Dec. 3 2015. US Patent App. 14/761,588

US 20150350278A1

(19) **United States**

(12) **Patent Application Publication**

Isbjörnssund et al.

(10) **Pub. No.: US 2015/0350278 A1**

(43) **Pub. Date: Dec. 3, 2015**

(54) **SECURE STREAMING METHOD IN A NUMERICALLY CONTROLLED MANUFACTURING SYSTEM, AND A SECURE NUMERICALLY CONTROLLED MANUFACTURING SYSTEM**

(71) Applicants: **Kimmo ISBJÖRNSSUND**, Tallinn (EE); **Anton VEDESHIN**, Tallinn (EE); **FABULONIA OÜ**, Tallinn (EE)

(72) Inventors: **Kimmo Isbjörnssund**, Tallinn (EE); **Anton Vedeshin**, Tallinn (EE)

(73) Assignee: **Trondert OÜ**, Tallinn (EE)

(21) Appl. No.: **14/761,588**

(22) PCT Filed: **Jan. 20, 2014**

(86) PCT No.: **PCT/EP2014/051065**

§ 371 (c)(1),
(2) Date: **Jul. 16, 2015**

(30) **Foreign Application Priority Data**

Jan. 19, 2013 (EP) 13151981.1
Jun. 7, 2013 (EP) 13171159.0

Publication Classification

(51) **Int. Cl.**

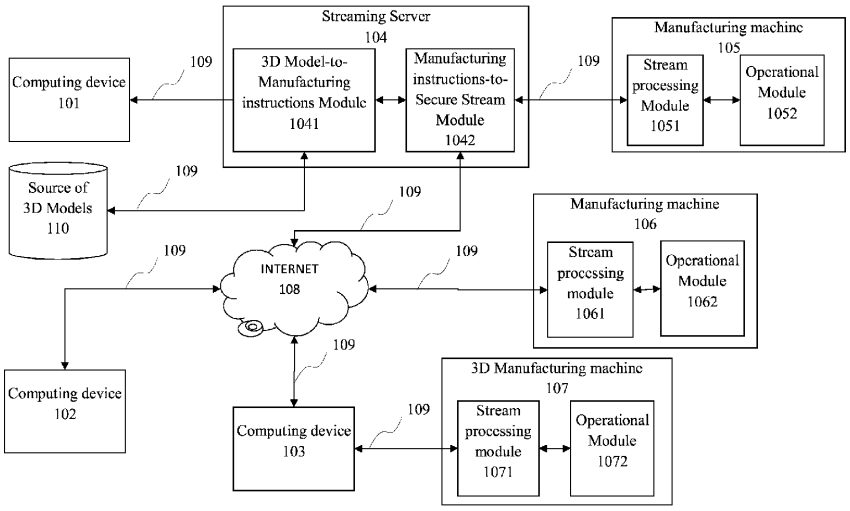
H04L 29/06 (2006.01)
B29C 67/00 (2006.01)
G05B 19/4099 (2006.01)

(52) **U.S. Cl.**

CPC ***H04L 65/60*** (2013.01); ***G05B 19/4099*** (2013.01); ***B29C 67/0088*** (2013.01); *B33Y 50/02* (2014.12)

(57) **ABSTRACT**

Secure streaming method in a numerically controlled manufacturing system, where the 3D file of the 3D object such as a CAD file or STL file is not sent to the manufacturing machine, but is kept in asecured system. Instead, only the instructions for controlling the manufacturing machine (e.g., so called G-codes) are streamed to the manufacturing machine. Such instructions are secured so that only a specific manufacturing machine can make use of them. To this end, the set of instructions may be encoded, e.g., hashed on a secure server, using a server hash table while the manufacturing machine is provided with a local lookup hash table that is synchronized, e.g., loosely synchronized with the serve's hash table for converting the hashed instructions back to instructions suitable for operating the manufacturing machine.



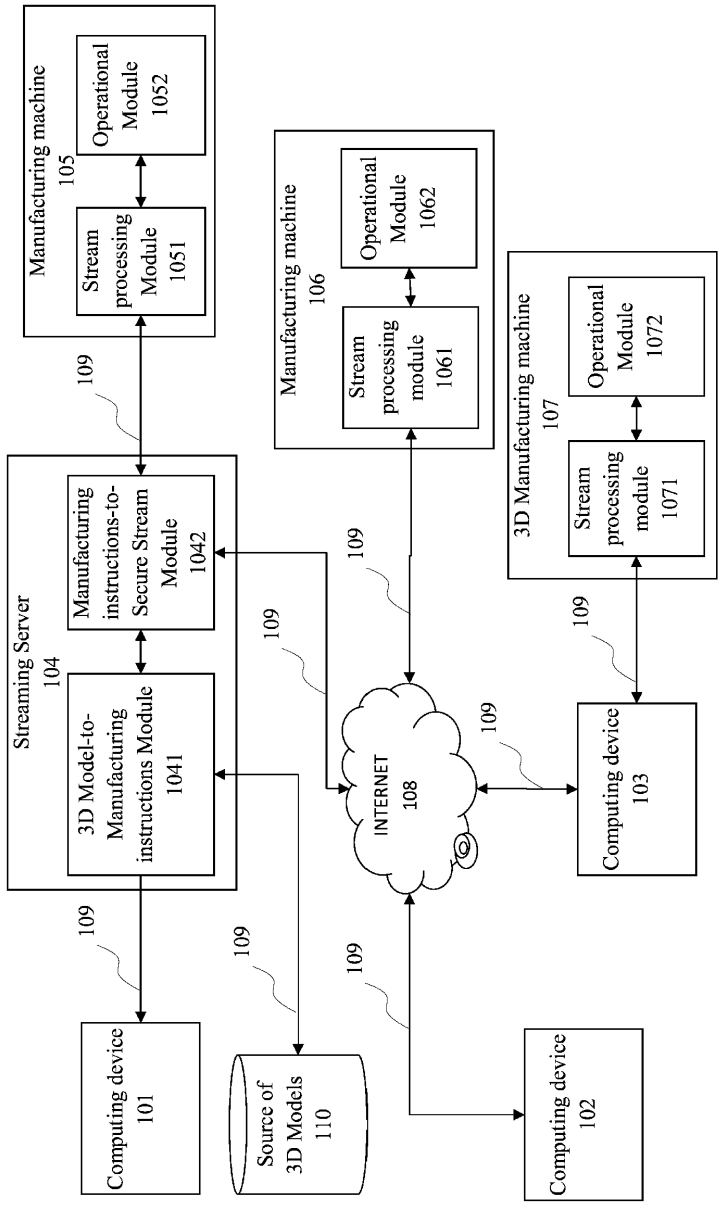

Manufacturing machine
105
Operational Module
1052
Stream processing Module
1051
Manufacturing machine
106
Operational Module
1062
Stream processing module
1061
3D Manufacturing machine
107
Operational Module
1072
Stream processing module
1071
Streaming Server
104
Manufacturing instructions-to-Secure Stream Module
1042
3D Model-to-Manufacturing instructions Module
1041
INTERNET
108
Computing device
103
Computing device
101
Source of 3D Models
110
Computing device
102
109

Fig. 1

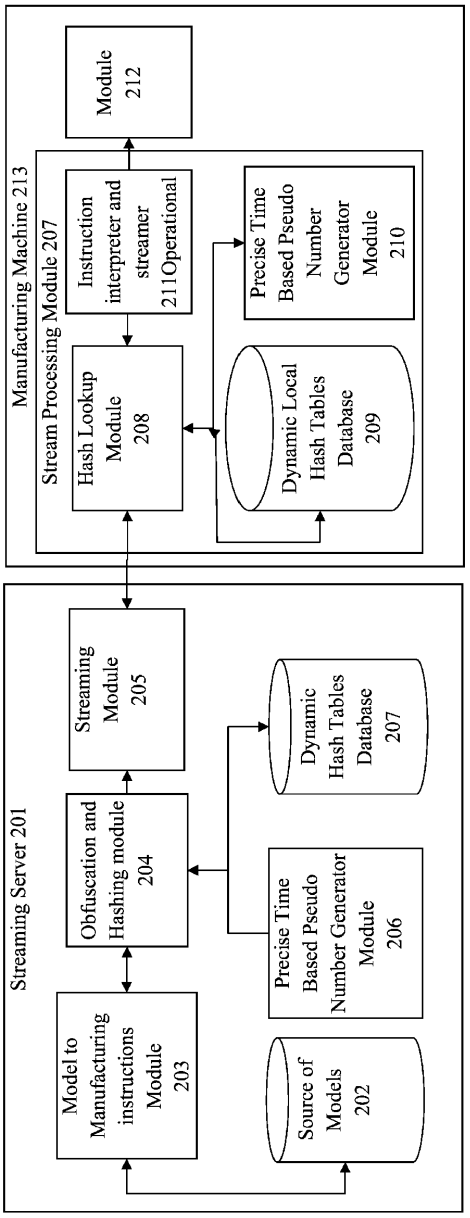

Streaming Server 201
Model to Manufacturing instructions Module 203
Source of Models 202
Obfuscation and Hashing module 204
Precise Time Based Pseudo Number Generator Module 206
Streaming Module 205
Dynamic Hash Tables Database 207
Manufacturing Machine 213
Stream Processing Module 207
Hash Lookup Module 208
Instruction interpreter and streamer 211Operational
Module 212
Dynamic Local Hash Tables Database 209
Precise Time Based Pseudo Number Generator Module 210

Fig. 2

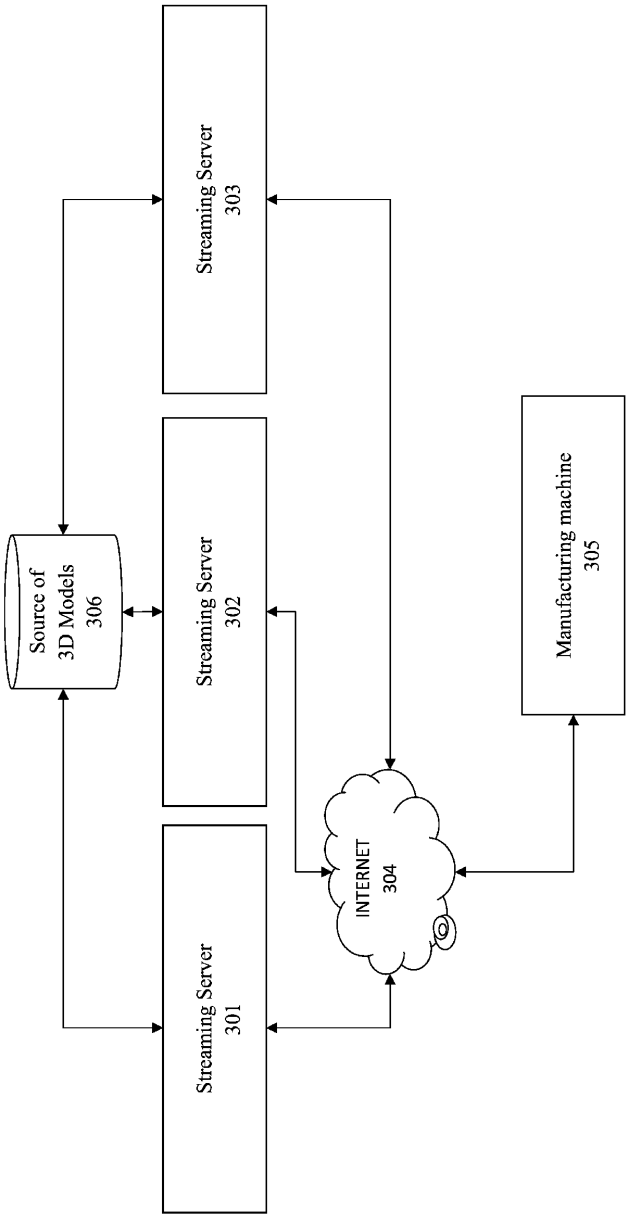

Streaming Server
303
Source of
3D Models
306
Streaming Server
302
Manufacturing machine
305
INTERNET
304
Streaming Server
301

Fig. 3

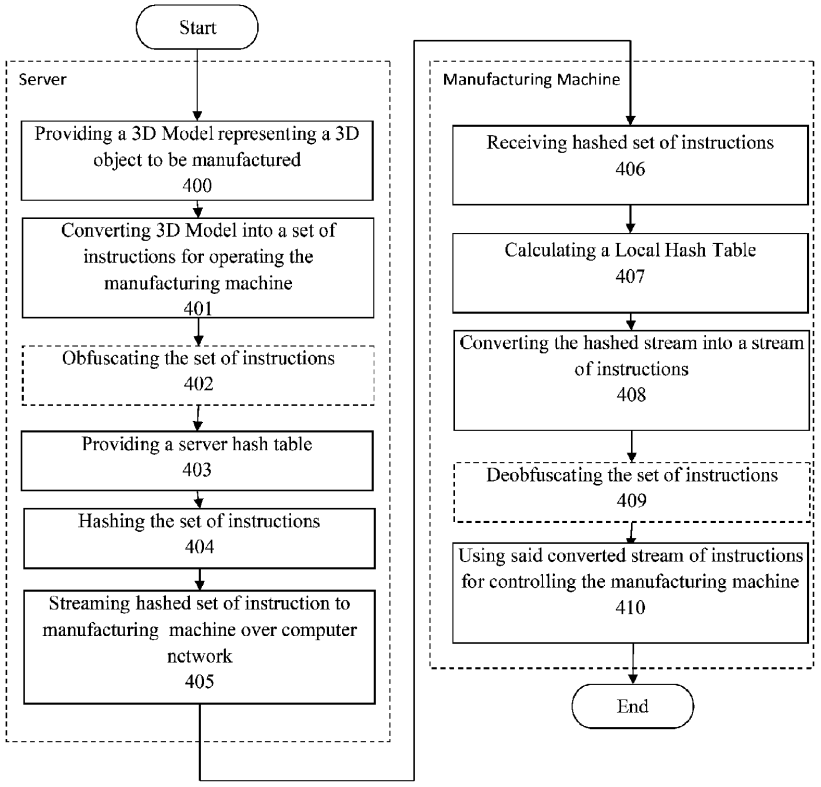

Start
Server
Providing a 3D Model representing a 3D object to be manufactured
400
Converting 3D Model into a set of instructions for operating the manufacturing machine
401
Obfuscating the set of instructions
402
Providing a server hash table
403
Hashing the set of instructions
404
Streaming hashed set of instruction to manufacturing machine over computer network
405
Manufacturing Machine
Receiving hashed set of instructions
406
Calculating a Local Hash Table
407
Converting the hashed stream into a stream of instructions
408
Deobfuscating the set of instructions
409
Using said converted stream of instructions for controlling the manufacturing machine
410
End

Fig. 4

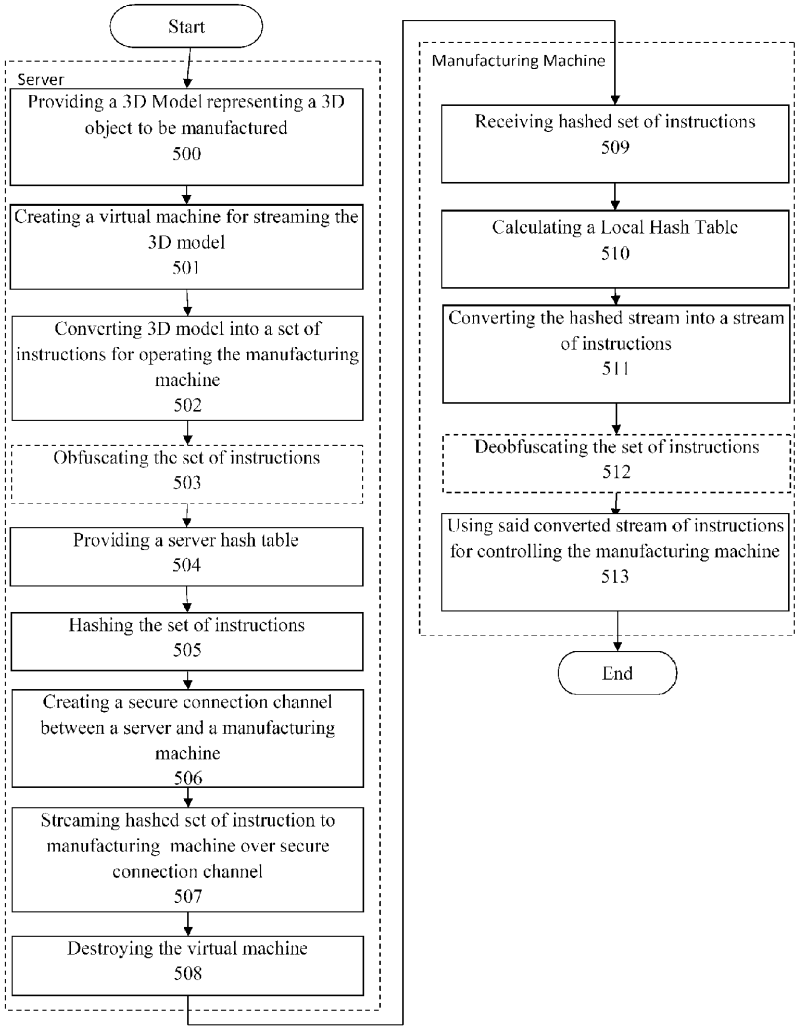

Start
Server
Providing a 3D Model representing a 3D object to be manufactured
500
Creating a virtual machine for streaming the 3D model
501
Converting 3D model into a set of instructions for operating the manufacturing machine
502
Obfuscating the set of instructions
503
Providing a server hash table
504
Hashing the set of instructions
505
Creating a secure connection channel between a server and a manufacturing machine
506
Streaming hashed set of instruction to manufacturing machine over secure connection channel
507
Destroying the virtual machine
508
Manufacturing Machine
Receiving hashed set of instructions
509
Calculating a Local Hash Table
510
Converting the hashed stream into a stream of instructions
511
Deobfuscating the set of instructions
512
Using said converted stream of instructions for controlling the manufacturing machine
513
End

Fig. 5

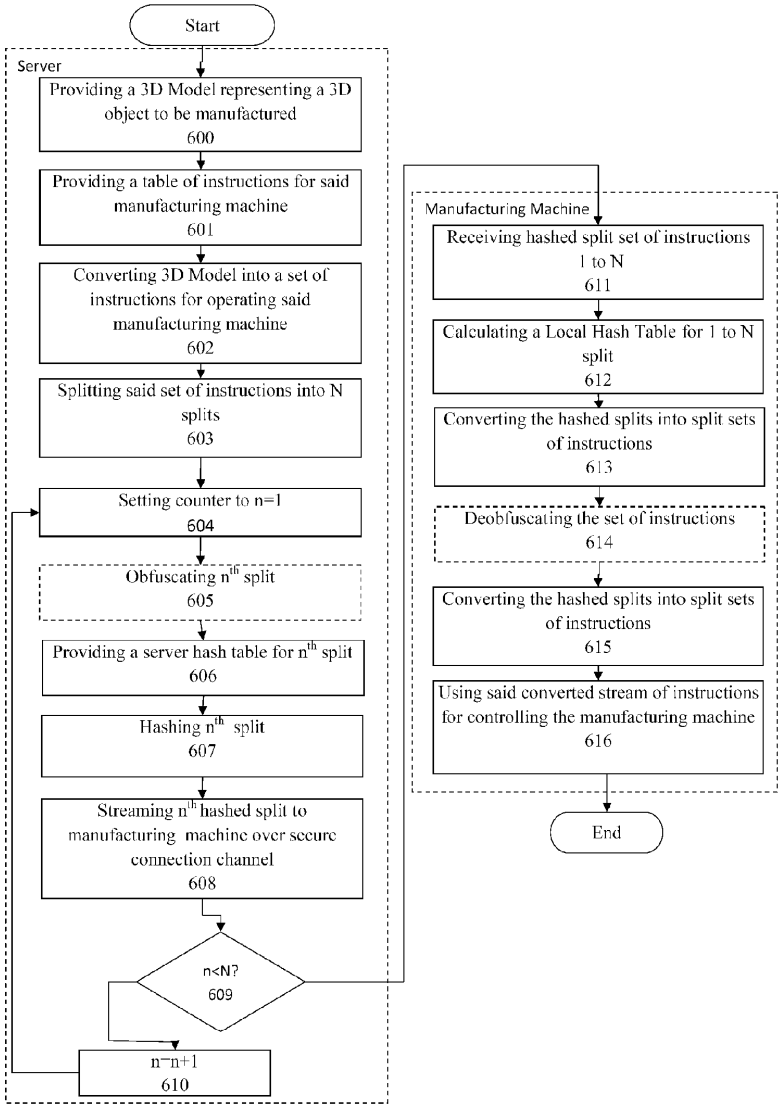

Start
Server
Providing a 3D Model representing a 3D object to be manufactured
600
Providing a table of instructions for said manufacturing machine
601
Converting 3D Model into a set of instructions for operating said manufacturing machine
602
Splitting said set of instructions into N splits
603
Setting counter to n=1
604
Obfuscating nth split
605
Providing a server hash table for nth split
606
Hashing nth split
607
Streaming nth hashed split to manufacturing machine over secure connection channel
608
n<N?
609
n=n+1
610
Manufacturing Machine
Receiving hashed split set of instructions 1 to N
611
Calculating a Local Hash Table for 1 to N split
612
Converting the hashed splits into split sets of instructions
613
Deobfuscating the set of instructions
614
Converting the hashed splits into split sets of instructions
615
Using said converted stream of instructions for controlling the manufacturing machine
616
End

Fig. 6

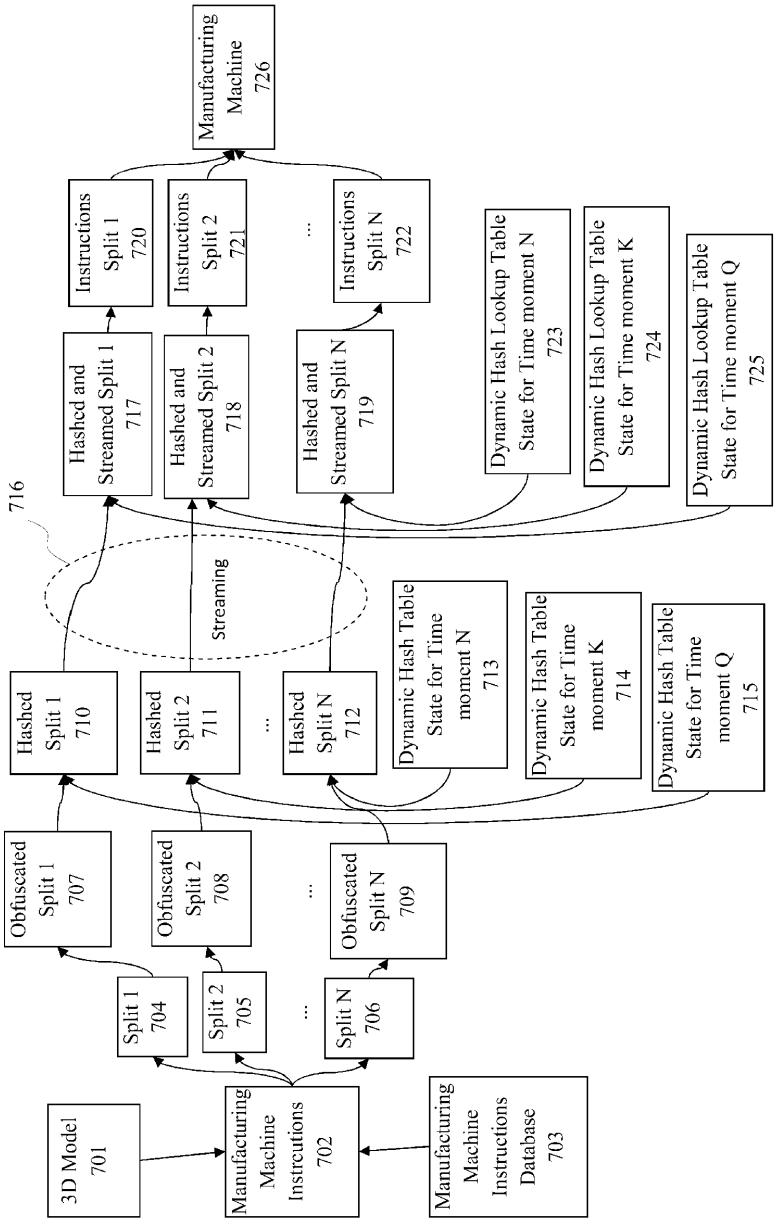

3D Model 701
Manufacturing Machine Instrcutions 702
Manufacturing Machine Instructions Database 703
Split 1 704
Split 2 705
Split N 706
Obfuscated Split 1 707
Obfuscated Split 2 708
Obfuscated Split N 709
Hashed Split 1 710
Hashed Split 2 711
Hashed Split N 712
Dynamic Hash Table State for Time moment N 713
Dynamic Hash Table State for Time moment K 714
Dynamic Hash Table State for Time moment Q 715
716
Streaming
Hashed and Streamed Split 1 717
Hashed and Streamed Split 2 718
Hashed and Streamed Split N 719
Instructions Split 1 720
Instructions Split 2 721
Instructions Split N 722
Dynamic Hash Lookup Table State for Time moment N 723
Dynamic Hash Lookup Table State for Time moment K 724
Dynamic Hash Lookup Table State for Time moment Q 725
Manufacturing Machine 726

Fig. 7

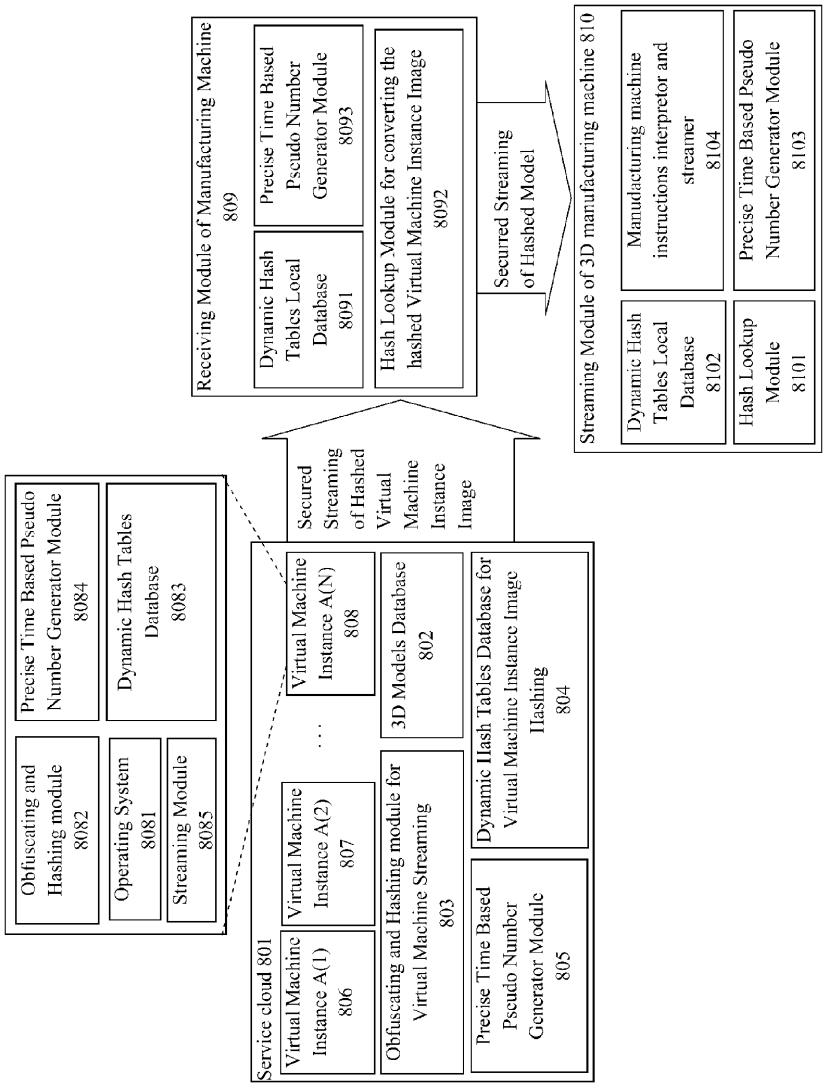

Obfuscating and Hashing module 8082
Precise Time Based Pseudo Number Generator Module 8084
Operating System 8081
Dynamic Hash Tables Database 8083
Streaming Module 8085
Service cloud 801
Virtual Machine Instance A(1) 806
Virtual Machine Instance A(2) 807
Virtual Machine Instance A(N) 808
Obfuscating and Hashing module for Virtual Machine Streaming 803
3D Models Database 802
Precise Time Based Pseudo Number Generator Module 805
Dynamic Hash Tables Database for Virtual Machine Instance Image Hashing 804
Secured Streaming of Hashed Virtual Machine Instance Image
Receiving Module of Manufacturing Machine 809
Dynamic Hash Tables Local Database 8091
Precise Time Based Pseudo Number Generator Module 8093
Hash Lookup Module for converting the hashed Virtual Machine Instance Image 8092
Securred Streaming of Hashed Model
Streaming Module of 3D manufacturing machine 810
Dynamic Hash Tables Local Database 8102
Manudacturing machine instructions interpretor and streamer 8104
Hash Lookup Module 8101
Precise Time Based Pseudo Number Generator Module 8103

Fig. 8

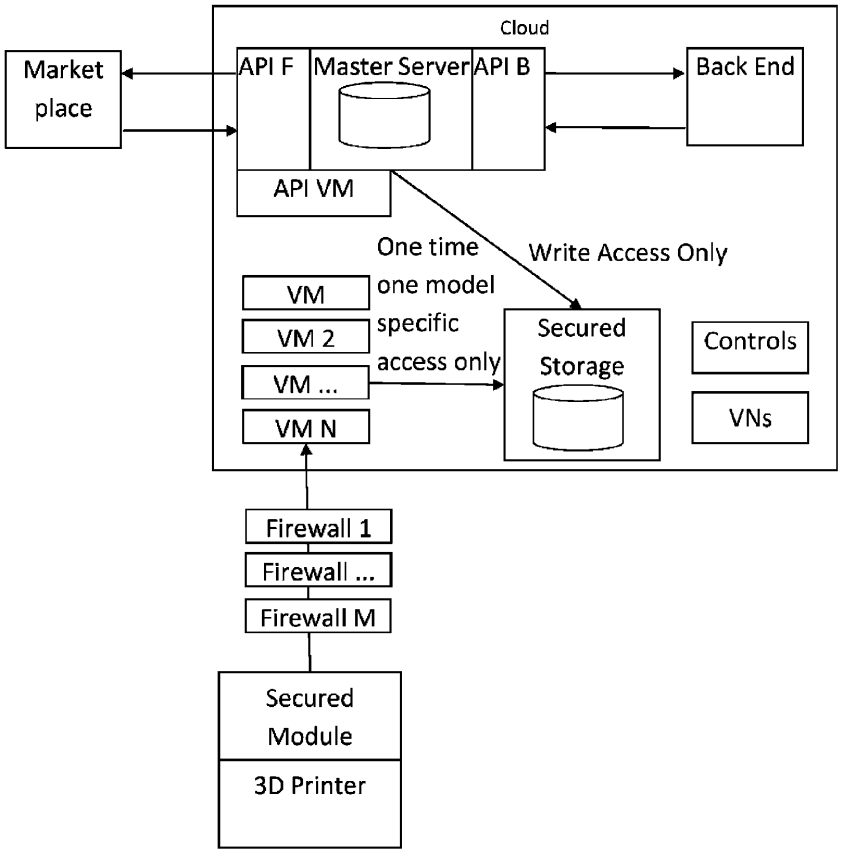

Cloud
Market place
API F
Master Server
API B
Back End
API VM
One time
Write Access Only
VM
VM 2
VM ...
VM N
one model specific access only
Secured Storage
Controls
VNs
Firewall 1
Firewall ...
Firewall M
Secured Module
3D Printer

Fig. 9

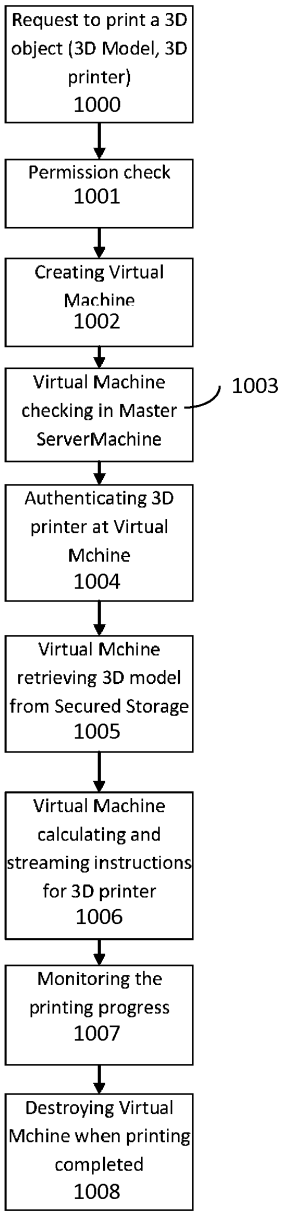

Request to print a 3D object (3D Model, 3D printer)
1000
Permission check
1001
Creating Virtual Machine
1002
Virtual Machine checking in Master ServerMachine
1003
Authenticating 3D printer at Virtual Mchine
1004
Virtual Mchine retrieving 3D model from Secured Storage
1005
Virtual Machine calculating and streaming instructions for 3D printer
1006
Monitoring the printing progress
1007
Destroying Virtual Mchine when printing completed
1008

Fig. 10

SECURE STREAMING METHOD IN A NUMERICALLY CONTROLLED MANUFACTURING SYSTEM, AND A SECURE NUMERICALLY CONTROLLED MANUFACTURING SYSTEM

TECHNICAL FIELD

[0001] The present invention relates to numerically controlled manufacturing systems, including rapid manufacturing and prototyping machines and systems, both by additive and subtractive methods, including 3D printing devices, with secure streaming of instructions for operating a manufacturing machine from a secure streaming server over a connection channel to a manufacturing machine, and more specifically, to methods and protocols used for streaming data in such systems.

BACKGROUND ART

[0002] Rapid manufacturing and rapid prototyping are relatively new class of technologies that can automatically construct physical 3D objects from Computer-Aided Design (CAD) data. Usually these methods make use of additive manufacturing technologies such as 3D printers.

[0003] 3D printing or additive manufacturing (AM) is a process of joining materials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies, such as traditional machining where the object is shaped by removing material. Several technologies are available for industrial uses, including for rapid prototyping and rapid manufacturing but increasingly so also for domestic and hobbyist uses. 3D printing is rapidly becoming as widespread as traditional 2D printing has become long ago.

[0004] Known is, e.g., WO2004/006087, disclosing a secure printing method in a traditional (2D) printing environment, where the print job as PDL print file such as PostScript file is encrypted with a cryptographic keys generated by the printer and then sent to the printer for decryption and printing the print job. While the method is useful to prohibit intercepting the print job by other devices in the network, this method does not avoid misuse of the print job by the printer itself and thus, leaves the owner of the rights of the document unprotected.

[0005] Combining 3D printing with 3D scanning makes possible 3D copying, i.e., a process where first a digital 3D model of an object is made by 3D scanning of the object and then a 3D copy of the 3D object is made by 3D reproducing the object similarly to the process of digital 2D copying.

[0006] It is well known that 2D printing and copying can be used to make copies of copyrighted materials or other materials protected by other types of intellectual property rights. While some technologies exist to inhibit copying, e.g., documents with security features such as watermarks, holograms, straps, UV or IR glowing, etc; however, no universally applicable technology exists to control reproducing and copying of copyrighted materials or other protected materials.

[0007] The problem becomes even more important in 3D printing and copying. For example, 3D objects can be subjects to different types of intellectual property rights independent from each other, including copyright (e.g., as sculptures, figurines, architectural objects, etc), industrial design (known in the US as design patent; e.g., a new shape of a product such as a vase or a chair), 3D trademark, by a patent (invention patent in the US) or a utility 3D model, or by personality rights (e.g., the likeness of a person). While certain fair use provisions may exist in copyright law (or analogous provisions for design patent or invention patent) allowing in some cases making copies for non-commercial private use, making copies of such 3D objects protected by intellectual property rights is prohibited at least for business purposes without a prior explicit permission (a license) from the right holder.

[0008] Known is U.S. Pat. No. 8,286,236 to Jung, titled Manufacturing control system, disclosing a method for secure manufacturing to control object production rights, such method comprises identifying at least one object data file configured to produce an object by a manufacturing machine; confirming that an authorization code is associated with the object data file, the authorization code configured to be received by the manufacturing machine, the manufacturing machine adapted to receive the authorization code; and enabling the manufacturing machine to interface with the object data file only if the authorization code meets one or more predetermined conditions, wherein the manufacturing machine is configured for at least one or more of additive manufacturing, subtractive manufacturing, extrusion manufacturing, melting manufacturing, solidification manufacturing, ejection manufacturing, die casting, or a stamping process. This approach is not secure enough as the 3D file can be freely copied and distributed and once the code is broken, the 3D file can be distributed without any control.

[0009] Known is WO2012/146943 to Within Technologies Ltd, titled Improvements for 3D design and manufacturing systems, disclosing a method of authenticating the printing of a three-dimensional (3D) article at a 3D printer according to an encrypted 3D print file describing a 3D design. The method comprises: receiving an authentication request from a 3D print server that is associated with the 3D printer, the request comprising a unique design identifier associated with a 3D design file and a unique 3D printer identifier associated with a 3D printer, the received unique 3D design identifier being related to the received 3D printer identifier in accordance with a first relationship; using at least one of the received unique identifiers to access a verifying 3D design identifier and a verifying 3D printer identifier, the verifying identifiers being related to each other in accordance with a second relationship; comparing the first and second relationships between the received and verifying identifiers; generating an authentication signal if the first relationship corresponds with the second relationship; obtaining a decryption key associated with the received identifiers in response to the authentication signal; and transferring the decryption key to the 3D print server to authenticate and enable the printing of the 3D article on the 3D printer. This solution may be considered as closest prior art.

[0010] Known methods are based on providing the 3D file with an authorization code or identifier for determining the authenticity of the 3D file. The use of the 3D file is controlled by the user right to access or print the 3D file. While these methods are suitable to inhibit unauthorized use of the 3D file itself, this approach is in fact misplaced as the object that is protected by copyright, design right or other intellectual property rights is not the file, but the 3D object itself While modifying the file can be perfectly legal, the prohibited activity is the unauthorized reproduction of the 3D object itself.

[0011] While it is important to allow users and manufacturers to determine if any restrictions exist on reproducing a 3D object, in preferred cases there must also be a mechanism

in place to actually prevent the unauthorized reproduction of the 3D object. As the 3D file itself representing the 3D object according to this scenario does not necessarily have any means attached preventing unauthorized use of the 3D file, the known methods cannot be used. The authorization means must be integrated with the manufacturing device itself, e.g., before each manufacturing work, the manufacturing device needs an authorization from the rights holder, or confirmation that no restrictions exist.

[0012] The method similar to WO2012/146943 could be used, i.e., all the 3D files could be received from and sent through a service provider who modifies the 3D files by encrypting the file and providing it with identification codes. However, even though the 3D files that are transmitted in the system are encrypted, they can be copied, saved, intercepted and thus, misused, e.g., by breaking the code and after that making the 3D files available in the Internet or through file sharing solutions. Therefore, more secure system is needed.

[0013] What is needed, therefore, is a more secure method and system where the 3D model of a 3D object is safe from unauthorized use, but the 3D object can nevertheless be manufactured in a numerically controlled manufacturing system.

SUMMARY OF INVENTION

[0014] The goal of the invention is achieved by a method and a system where the original 3D file of the 3D object such as a CAD file or STL file is not sent to the manufacturing machine, but is kept in a secured system and instead, only the instructions for controlling the manufacturing machine (e.g., so called G-codes) that are specific to this manufacturing machine are streamed to the manufacturing machine. Furthermore, such instructions are secured so that only a specific manufacturing machine can make use of them. Such manufacturing machine must be equipped with means for processing or converting said instructions into a format suitable for operating said manufacturing machine. To this end, the set of instructions may be encoded, e.g., hashed on a secure server, using a server hash table while the manufacturing machine is provided with a local lookup hash table that is synchronized, e.g., loosely synchronized with the server's hash table for converting the hashed instructions back to instructions suitable for operating the manufacturing machine. For example, time based or some common event or action based loose synchronization can be used.

[0015] According to one embodiment of the invention, a streaming method in a secure manufacturing system which comprises a streaming server and a numerically controlled manufacturing machine connected to said streaming server over a communication channel, the method comprises the steps of providing to the streaming server a model of a 3D object to be manufactured (hereinafter: 3D model) by said manufacturing machine, on said streaming server, converting said 3D model into a set of instructions for operating said manufacturing machine; encoding said set of instructions into a set of encoded instructions by applying simultaneously or in sequence one or more processes such as calculating a set of hashed instructions by applying a cryptographic hash function to said set of instructions, calculating a set of obfuscated instructions by applying obfuscation function to said set of instructions, applying arithmetic coding to said set of instructions, applying digital fingerprints, calculating checksums, calculating hash values, calculating digital DNA, and encrypting said set of instructions; and outputting said set of instructions to said manufacturing machine over said communication channel.

[0016] 3D models secured streaming algorithm is using one way functions, i.e., functions that produce easy to compute strings for any given streaming block, but from these strings it is not possible to generate initial block. Also, it is impossible to modify the initial block without modifying said string. Moreover it is infeasible to find two different blocks which correspond to the same generated string. The cryptographic hash functions include such well known functions such as message digest algorithms (MD4, MD5), secure hash algorithms (SHA-1, SHA-2, SHA-3), Skein, Keccak, RadioGatun, PANAMA, and many others. The ideal cryptographic hash function has four main properties: it is easy to compute the hash value for any given message; it is infeasible to generate a message that has a given hash; it is infeasible to modify a message without changing the hash; it is infeasible to find two different messages with the same hash. Instead of cryptographic hash functions, non-cryptographic hash functions can be used as well as other one way functions having similar properties (i.e., easy to compute on every input, but hard to invert given the image of a random input) can be used for hashing. Even though general purpose hash functions can be used, also special purpose hash function can be designed, taking into account the nature of the data to be hashed (i.e., the instructions for controlling the manufacturing machine). Checksum functions, cyclic redundancy checks, checksums and fingerprinting functions can be used for hashing. Hashing can be performed using nonlinear table lookup.

[0017] According to another embodiment, on said streaming server a server hash table is generated; said set of instructions are hashed into a hashed set of instructions, using said server hash table; and the hashed set of instructions are outputted as a hashed stream of instructions to said manufacturing machine over said communication channel. On the manufacturing machine side, the hashed stream is received, a local hash table corresponding to and synchronized, e.g., loosely synchronized (e.g., time-based, action based) to said server hash table is calculated on said manufacturing machine, the hashed stream is converted to a stream of instructions, using said local hash table into and the converted stream of instructions is used to operate the operational part of the manufacturing machine.

[0018] According to one embodiment, the method comprises during said hashing periodically regenerating said hash table and correspondingly regenerating said local hash table during said converting said hashed stream according to a first predetermined precise time algorithm or other algorithm based on action or happening which are known to both the streaming server and a manufacturing machine independently, without actual sending or receiving information between each other.

[0019] According to one embodiment, the method additionally comprises splitting said set of instructions into split sets of instructions, obfuscating each of said split sets of instructions, hashing each of said obfuscated splits, streaming said hashed obfuscated splits independently over said communication channel from the streaming server to the manufacturing machine, converting said streamed splits into split sets of instructions and combining said split sets of instructions into the stream of instructions for controlling the manufacturing machine.

[0020] According to one embodiment, providing said 3D model comprises creating a secure connection over a communication channel between the streaming server and a source of 3D models, hashing said 3D model at the source of 3D models, transferring said hashed 3D model to said streaming server, before and re-hashing said hashed 3D model for streaming to said manufacturing machine.

[0021] According to one embodiment of the invention, the virtual machine is created and destroyed for each instance of streaming. Destroying of the virtual machine after the streaming is completed provides higher security as the server hash table cannot be recovered or reused.

[0022] According to one embodiment, the method additionally comprises destroying said virtual machine and creating new virtual machine instance so that each instance of streaming is carried out by more than one virtual machine.

[0023] According to one embodiment, the method additionally comprises creating more than one virtual machine for each instance of streaming, so that different parts of said 3D model are streamed by different virtual machines.

[0024] According to one embodiment, the system further comprises a computer device with a source of 3D models and the computer device is connected to said streaming server over a communication channel, and the method further comprises the steps of creating on said computer device a first virtual machine for providing said 3D model to said streaming server, hashing said 3D model in said first virtual machine, creating a secured virtual machine instance on said streaming server, receiving hashed 3D model by said secured virtual machine instance, storing said hashed 3D model in memory hash table, materializing said secured virtual machine instance into hashed virtual machine instance image, said image is transferred to a second computer device connected to a manufacturing machine, running said secured virtual machine instance on said second computer device and streaming locally said hashes of the 3D model to said manufacturing machine.

[0025] According to one embodiment, the secure manufacturing system comprises a plurality of streaming servers. Each streaming server is connected to the Internet and said steps of secure streaming are carried out by more than one streaming server in concert. Each of said streaming servers may be set up to stream a different part of said 3D model to be manufactured.

[0026] The goals of the invention are also achieved by a secure numerically controlled manufacturing system, the system comprising a streaming server, having a conversion module adapted for receiving a 3D model representing a 3D object to be manufactured and for converting said 3D model into a set of manufacturing instructions, an obfuscating and hashing module adapted to obfuscate and to hash said set of manufacturing instructions into a hashed set of instructions, a dynamic hash tables database adapted to provide hash tables for said hashing module and a precise time based pseudo number generator module; a source of 3D models, connected to said streaming server over a communication channel; and a manufacturing machine, connected to said streaming server over a communication channel, said manufacturing machine comprising an operational module, a hash lookup module for converting said hashed set of instructions, a Dynamic Local Hash Tables Database for providing hash tables for hash lookup module and precise time based pseudo number generator module for independently synchronizing the hash tables of the manufacturing machine with the hash tables used on said streaming server. The system may comprise a plurality of streaming servers, each of said streaming servers connected to the Internet and adapted perform said secure streaming in concert.

[0027] The system according to one embodiment comprises a 3D printer equipped with a secured module and having a connection to a Cloud; a Master Server located in the Cloud, said Master Server comprising a front-end application programming interface for Front End API F and an application programming interface for the back end API B. Marketplaces such as web stores providing 3D models are connected to the Master Server through the API F. 3D models can be uploaded to the system into a Secure Storage in the Cloud using back end through the API B.

[0028] The system is operated as follows. The 3D objects offered for reproduction are shown on the Marketplaces (preferably as 2D images, i.e., not the actual 3D model files). The user picks a specific 3D object to be reproduced, and indicates a specific 3D printer to be used (e.g., the one connected to her computer over USB port). Upon receiving a request from the user, the Master Server first checks the permission to reproduce the 3D object and then creates a Virtual Machine for securely streaming instructions necessary for reproducing the 3D object to the 3D printer. Such Virtual Machine is created only for streaming one specific 3D model and to only one specific 3D printer. The Virtual Machine (and only the Virtual Machine) can access the Secure Storage to access this specific 3D model. Only one specific 3D printer is associated with and can access one Virtual Machine. The 3D printer connects to the Virtual Machine as follows. When the 3D printer is connected to the network, it connects to the Master Server using personal certificate. Secure channel is then established between 3D printer and the Master Server when the 3D printer is plugged into the network.

[0029] When the Virtual Machine is created, the Master Server provides the Virtual Machine with an IP address and port number. The 3D printer is associated with the IP address and port and creates secure network with the Virtual Machine, using, e.g., Virtual Private Network (VPN). The connection is possible only if the personal certification matches the certificate on Virtual Machine.

[0030] The streaming protocol includes:

[0031] Authorization. Virtual Machine is checking from the Master Server whether the permission exists to print 3D model.

[0032] Network speed check (e.g., the Virtual Machine sends one file of sufficient size and determines the time spent, and the 3D printer sends another file); if the speed is good enough, the secure streaming can begin. Speed check can be repeated during the printing process; printing can be resumed in case of network interruptions.

[0033] Hashing a set of G-codes into one block, and sending the blocks. When the block is sent, the Virtual Machine communicates to the Master Server the status update.

[0034] After the 3D model is reproduced, the Virtual Machine is destroyed.

[0035] More than one Virtual Machines can be created for printing single 3D object for increased security. For example, first Virtual Machine is created and streams first portion of the 3D object. Then the first Virtual Machine is destroyed, the Second Virtual Machine is created and streams the second portion of the 3D object, and so on until the 3D object is finished. Then the last Virtual Machine is destroyed.

[0036] The invention is also the method as shown in FIG. **10**.

BRIEF DESCRIPTION OF DRAWINGS

[0037] FIG. **1** is a block diagram of exemplary system that supports the claimed subject matter of the present application.

[0038] FIG. **2** is a block diagram of one embodiment of the secure streaming server and stream processing module of the manufacturing machine.

[0039] FIG. **3** is a block diagram of a multimode streaming system.

[0040] FIG. **4** is a flow chart of a method according to one embodiment of the invention.

[0041] FIG. **5** is a flow chart of a method according to another embodiment of the invention.

[0042] FIG. **6** is a flow chart of a method according to another embodiment of the invention.

[0043] FIG. **7** is a block diagram explaining a method according to still another embodiment of the invention.

[0044] FIG. **8** is a block diagram of the system according to one embodiment of the invention.

[0045] FIG. **9** depicts a block diagram of a system according to one embodiment of the present invention.

[0046] FIG. **10** depicts a flow diagram of a method according to one embodiment of the present invention.

DESCRIPTION OF EMBODIMENTS

[0047] Definitions

[0048] 3D printer means any device suitable for making a three-dimensional solid object of virtually any shape from a 3D digital model.

[0049] 3D printing means any numerically controlled automated manufacturing process. Cloud (or, a Computing Cloud) describes a variety of different computing concepts that involve a large number of computers that are connected through a real-time communication network (typically, the Internet).

[0050] The block diagram of exemplary system that supports the claimed subject matter of this patent application is shown on FIG. **1**. The system comprises one or more computing devices **101**, **102** and **103** that are connected to Streaming Server **104** over a communication channel **109**, including the Internet **108**. The Streaming Server has one or more Manufacturing Machines **105**, **106** and **107** such as 3D printers, etc, connected to it over a communication channel **109**. The system also comprises a source of 3D models **110** for providing 3D models for the streaming server. The connection between the Streaming Server **104** and manufacturing machines is preferably over a secured channel, such as TLS and SSL for the Internet. The Streaming Server comprises a module **1041** for converting 3D models into a set of manufacturing instructions and a module **1042** for converting said set of instructions into a set of encoded instructions. The manufacturing machine comprises a module for stream processing (**1051**, **1061** and **1071**, correspondingly) and an operational module (**1052**, **1062** and **1072**, correspondingly) responsible for manufacturing the 3D object.

[0051] The 3D model here is any computer model of a 3D object to be manufactured, such as file(s) in any of the computer aided design (CAD) file format, STL file(s), or additive manufacturing file format. It can also be one or more files providing views of the 3D object in any image file format.

[0052] The manufacturing machine can be any numerically controlled manufacturing machine, such as three-dimensional additive manufacturing machines configured for rapid prototyping, three-dimensional printing, two-dimensional printing, freeform fabrication, solid freeform fabrication, and stereolithography. Manufacturing machines can also include a subtractive manufacturing machine, including machines adapted for drilling, milling, turning, laser cutting, waterjet cutting, plasma cutting, wire electrical discharge cutting, cold, warm and hot forging metal fabrication, computer numerical controlled fabrication machine, and/or an additive manufacturing machine, and/or an injection molding machine. The manufacturing machines further include an extrusion manufacturing machine, a melting manufacturing machine, a solidification manufacturing machine, an ejection manufacturing machine, a die casting manufacturing machine, a stamping process machine, an assembly robot assembling 3D objects from pieces or blocks.

[0053] The manufacturing machines can include a manufacturing machine configured to perform manufacturing using one or more of metal, wood, ice, stone, glass, nuclear materials, pharmaceuticals, edible substances, living substances, cells, chemical molecules, sand, ceramic materials, aluminium, silicon, carbides, silicon nitrides, silicon carbides, metal/ceramic combinations including aluminium/silicon nitride, aluminium/silicon carbide, aluminium/zirconium and aluminium/aluminium nitride including materials alterable by friction, heating and cooling.

[0054] The manufacturing instructions can be, e.g., G-codes or other instructions according to any computer language, including numerical control (CNC) programming language, but also high-level languages like python, java, PHP, etc. Such manufacturing instructions define where to move to, how fast to move, and through what path to move the operative part of the manufacturing machine, such as the printing head, the extruder head, etc, as well as other manufacturing parameters.

[0055] The communication channel can be provided by any technology used for numerically controlling manufacturing machines, e.g., any computer network using any communication media (i.e., wireless or wired), communication protocol (e.g., Internet Protocol, or Ethernet protocol, etc), or scale (e.g., near field network, personal network, local area network, wide area network. Also virtual private networks, peer to peer connections, or over satellite communication channels may be used.

[0056] The block diagram shown on FIG. **2** further clarifies the architecture of the streaming server **201** according to one embodiment and corresponding manufacturing machine **213** comprising a Stream Receiving Module **207** and an Operational Module **212**. The Streaming server **201** according to this embodiment comprises a Source of 3D models **202** for providing 3D models, a module **203** for converting 3D model to manufacturing instructions, a module **204** for obfuscating and hashing the manufacturing instructions into a hashed stream, and a Streaming Module **205** for outputting said hashed stream over a computer network to the manufacturing machine. The hashing is controlled by Precise Time Based Pseudo Number Generator Module and performed using a hash table provided by a Dynamic Hash Tables Database **207**.

[0057] The stream processing module **207** comprises a Hash Lookup Module **208** for converting the hashed stream into stream of instructions. This converting is controlled by Precise Time Based Pseudo Number Generator Module **210**

and performed using a Dynamic Local Hash Tables Database **209**. The converted stream of instructions is sent to the operational module using instruction interpreter and streamer **211**.

[0058] The block diagram of FIG. **3** shows a multimode streaming system, comprising several Secure 3D Object Streaming Servers (shown as **301**, **302** and **303**), connected to computer network such as Internet **304**, a manufacturing machine **305**, also connected to the computer network, and at least one source of 3D models **306** for providing 3D models to be streamed.

[0059] One embodiment of the secure streaming method is shown as a flowchart in FIG. **4**. The secure streaming method comprises the steps of providing a 3D model representing a 3D object to be reproduced **400**, converting said 3D model into a set of instructions, such as G-codes for operating the manufacturing machine **401**, optionally obfuscating said set of instructions **402**; providing a server hash table **403**, hashing said set of instructions **404** and streaming said hashed set of instructions to manufacturing machine over a communication channel **405**. On the manufacturing machine side, the method comprises the steps of receiving the hashed set of instructions **406**, calculating on said manufacturing machine a Local Hash Table corresponding to and loosely synchronized to said server hash table **407**, converting the hashed stream into a stream of instructions, using said Local Hash Table **408**, deobfuscating the stream of instructions, if necessary **409** and using the converted stream of instructions for controlling the operational part of the manufacturing machine **410**.

[0060] The flow diagram of FIG. **5** shows a modified embodiment of the invention. The secure streaming method comprises the steps of providing a 3D model representing a 3D object to be reproduced **500**, creating a virtual machine for streaming the 3D model **501**, converting said 3D model into a set of instructions, such as G-codes for operating the manufacturing machine **502**, optionally obfuscating said set of instructions **503**; providing a server hash table **504**, hashing said set of instructions **505**, creating a secure connection channel between a server and a manufacturing machine **506**, streaming said hashed set of instruction to manufacturing machine over secure connection channel **507** and destroying the virtual machine **508**. This approach makes it impossible to recover the hash table used for hashing from the server side as it is permanently destroyed together with the virtual machine. On the manufacturing machine side, the method comprises the steps of receiving the hashed set of instructions **509**, calculating on said manufacturing machine a Local Hash Table corresponding to and loosely synchronized to said server hash table **510**, converting the hashed stream into a stream of instructions, using said Local Hash Table **511**, deobfuscating the stream of instructions, if necessary **512** and using the converted stream of instructions for controlling the operational part of the manufacturing machine **513**.

[0061] The flow diagram of FIG. **6** shows another modified method. The secure streaming method comprises providing a 3D model representing a 3D object to be reproduced by a manufacturing machine **600**; providing a table of instructions for said manufacturing machine **601**; converting 3D model into a set of instructions for operating said manufacturing machine **602**; splitting said set of instructions into N splits **603**, setting a counter to one **604**; optionally obfuscating n^{th} split **605**, providing a server hash table for n^{th} split **606**; hashing n^{th} obfuscated split **607**; streaming n^{th} hashed set of instructions to manufacturing machine over secure connection channel **608**, checking if further splits exist **609**, and if so, repeating steps **605** to **608** for $n=(n+1)^{th}$ split **610**. This method provides increased security as several hash tables are used for hashing the same stream. On the manufacturing machine side, the method comprises the steps of receiving hashed split sets of instructions 1 to N **611**, calculating Local Hash Table for each 1 to N hashed split corresponding to and loosely synchronized to corresponding n^{th} server hash table **612**, converting said streamed hashed splits into split sets of instructions **613**, deobfuscating the split sets of instructions, if necessary **614**, combining said split sets of instructions into the stream of instructions for controlling the manufacturing machine **615** and using the converted stream of instructions for controlling the operational part of the manufacturing machine **616**.

[0062] Method as shown on FIG. **5** can be combined with the method as shown on FIG. **6**, i.e., by creating a virtual machine for obfuscating, hashing and streaming each n^{th} split and destroying the virtual machine as soon as the streaming of the n^{th} split is completed.

[0063] FIG. **7** shows a block diagram of another embodiment. 3D model **701** is provided. Manufacturing Machine Instructions **702** are calculated, using Manufacturing Machine Instructions Database **703**. The instructions are split into N splits shown as **704** to **706**. Then, the splits **704** to **706** are processed in parallel by first obfuscating the splits into obfuscated splits **707** to **709**, then hashing each of said obfuscated splits into hashed splits **710** to **712**, using a Dynamic Hash Table State for Time moment N **713**, a Dynamic Hash Table State for Time moment K **714**, and a Dynamic Hash Table State for Time moment Q **715** correspondingly. Each of the hashed splits **710** to **712** are then independently streamed over a network **716** Time moments N, Q and K may be unrelated to the specific split to be processed, so one dynamic hash table can be used to process more than one split, as well as more than one dynamic hash table can be used to process a single split.

[0064] At the receiving side, at the manufacturing machine, each of the hashed and streamed splits **717** to **719** are converted back to instructions splits **720** to **722**, using a Dynamic Hash Lookup Table State for Time Moment N **723**, a Dynamic Hash Lookup Table State for Time Moment N **724** and a Dynamic Hash Lookup Table State for Time Moment N **725** respectively, the splits are combined and outputted to the operational part of the Manufacturing Machine **726**.

[0065] FIG. **8** shows another embodiment of the invention. The server is run in a service cloud. The server comprises 3D models Database **802**, Obfuscating and Hashing Module for Virtual Machine Streaming **803**, A Dynamic Hash Tables Database for Virtual Machine Instance Image Hashing **804** and a Precise Time Based Pseudo Number Generator Module **805**. Several virtual machine Instances A(1) to A(N) (shown as **806** to **808**) can be initiated at the server, each virtual machine instance comprising an operating system **8081**, obfuscating and hashing module **8082**, a dynamic hash tables database **8083**, a precise time based pseudo number generator module **8084** and a streaming module **8085**. The hashed virtual machine instance image is streamed to the receiving module of manufacturing machine **809**, said module comprising a Dynamic Local Hash Tables Database **8091**, Hash Lookup Module for converting the Hashed Virtual Machine Instance image **8092** and precise time based pseudo number generator module **8093**. The hashed 3D model is then securely streamed to be converted to the stream of instruc-

tions principally as described above, using a Streaming module of the manufacturing machine **810**, comprising a Hash Lookup Module **8101**, a Dynamic Local Hash Tables Database **8102**, precise time based pseudo number generator module **8103** and Manufacturing machine instructions interpreter and streamer **8104**.

[0066] It is obvious for the skilled person that the different examples of the methods as described above can be freely combined. Similarly, the different examples of the systems as described can be freely combined. For example, instead of or in addition to hashing, other methods of encoding can be used, e.g. obfuscating the instructions, applying arithmetic coding to the instructions, or encrypting the instructions. Virtual Machines can be run in a cloud system. The streaming can be provided as a service in a cloud system. Each computing device connected to the network can be provided with software to run as a secure streaming server, so the designers can provide secure streaming of their 3D models for manufacturing. In a peer to peer system, each computing device connected to the peer to peer network can be programmed to act as a secure streaming server. Each computing device connected to the computer network, including the peer to peer network can be modified to act as a source of 3D models. Such computing device may be adapted to securely stream the 3D models to another secure streaming server for streaming to the manufacturing machine, or the source of 3D models can be integrated with secure streaming server to directly stream to the manufacturing machine.

[0067] The cryptographic hash functions include such well known functions such as message digest algorithms (MD4, MD5), secure hash algorithms (SHA-1, SHA-2, SHA-3), Skein, Keccak, RadioGatun, PANAMA, and many others. The ideal cryptographic hash function has four main properties: it is easy to compute the hash value for any given message; it is infeasible to generate a message that has a given hash; it is infeasible to modify a message without changing the hash; it is infeasible to find two different messages with the same hash. Instead of cryptographic hash functions, other one way functions having similar properties (i.e., easy to compute on every input, but hard to invert given the image of a random input) can be used for hashing. Even though general purpose hash functions can be used, also special purpose hash function can be designed, taking into account the nature of the data to be hashed (i.e., the instructions for controlling the manufacturing machine). Checksum functions, cyclic redundancy checks, checksums and fingerprinting functions can be used for hashing. Hashing can be performed using nonlinear table lookup.

[0068] The method and the system for secure streaming may be also useful in other fields of technology where secure streaming is required, e.g., 1. for streaming control commands for controlling objects from a distance, or 2. for streaming commands from one operating module to another module of a car, aircraft, ship, electronic or computing device, etc. 3. for media broadcasting (radio, television), 4. for broadcasting of 3D object from storage module to a presenting module of 3D device, like 3D projectors in 3D cinema, 3D TV, SMART TV, 3D gaming consoles, 3D mobile Apps, 3D virtual reality glasses, augmented reality applications and devices, 3D hologram devices and applications. It is immediately apparent for the skilled person that in this case, instead of instructions for controlling the manufacturing machine, different types of instructions, suitable for controlling such device need to be used.

[0069] While the method is based on streaming the instructions to the manufacturing machine, it could also include temporarily buffering or caching the stream in the manufacturing machine or on the server side before sending.

[0070] The system is shown on FIG. **1**. In the Cloud, there is a Master server comprising:

[0071] An API F (Application programming interface for Front End), which is preferably a secured API (for example SSL, other kind), used by a Marketplace of 3D models. The secured streaming is initialized through the Marketplace.

[0072] An API B (Application programming interface for Back End), which is preferably a secured API (for example SSL, other kind), used by back end solutions of right holders to securely upload 3D models into a Secured Storage of 3D object models.

[0073] An API VM (Application programming interface on Virtual Machines), which is preferably a secured API for communication with the Secured Storage of 3D object models.

[0074] Virtual Machines, wherein every virtual machine VM 1 to VM N instance is executed for predetermined amount of time, for specific (i.e., one and only) 3D object model to be reproduced and for specific (i.e., one and only) 3D printer to be used for such reproduction. After the streaming session is completed, the Virtual Machine responsible for this streaming session is destroyed. Streaming session uses floating hashing tables to secure the streaming process; using hash tables for secure streaming is described in co-pending EP application No EP13151981.1.

[0075] An authorization table for 3D printers is kept on Master Server. Such table contains information on registered 3D printers, unique printer identifiers, permissions (e.g., license) start and end date, time of streamed 3D models, current state of the registered 3D printer (busy, available, not connected, network error, etc.), etc.

[0076] The Cloud also comprises a Secured storage of 3D files, where the 3D files and their parameters, as well as the meta information is stored. The Master Server can access the Secured Storage only for writing (Write Access Only). Only the correct Virtual Machine can access the Secured Storage for reading 3D files from the Secured Storage.

[0077] Different parts of the system in the Cloud (the Master Server, the Virtual Machines, the front end, the back end, the Secured Storage, the 3D printers, etc) are connected to each other in using secured connection, such as virtual networks, such as OpenVPN.

[0078] There is a proprietary protocol used by different parts of the cloud for communicating to each other. This protocol utilizes hashing and other encryption algorithms.

[0079] A 3D printer is connectable to the Master server. 3D printer could be any kind of 3D printer (USB connected, networked, WiFi printer, etc.). The printer communicates with the Cloud through a chip inside the 3D printer, a board inside the printer, or through a standalone device connected to the printer, or using computer software outside of the printer. Both 3D Printer internal parts, and external parts could be physically secured by a silicon/other material solid filling, or metal in-casing to make it rather impossible to disassemble, or when disassembled, the device will become non-operative.

[0080] 3D printer is visible to a Cloud even if it is a behind a number of firewalls. 3D printer could have external IP address, but not necessarily. This is accomplished by so-called printer to server for virtual machine peer-to-peer virtual network.

[0081] The Master Server is adapted to run a number of detective checks which detect that if some suspicious activity happens in protocol, virtual network, cloud, master server, 3d printer, secure storage, virtual machine, etc., including ports scanning, excessive IP addresses in virtual network, wrong requests to API, behaviour inside protocol, alarm on every server (special commands and codes that should be executed in the first X seconds after connection to the server, port knocking before connection to the machines)

[0082] The secured 3D Printing Protocol used for secure streaming has the following parts:

[0083] Establishing a secured connection between the 3D printer and corresponding Virtual Machine, using two way SSL certificates;

[0084] Authorizing the 3D printer using personal certificates, unique identification number, etc.

[0085] Checking Network quality and speed (using, e.g., ping, upstream, downstream).

[0086] Sending blocks of hashed and preferably crypted g-codes, STL file chunks, etc.

[0087] Controlling the printing process (pause, stop, resume, status, temperature of extruders, etc.)

[0088] Checking the quality of the 3D printing, e.g., by providing video or photo stream of the printed model.

[0089] Marketplace could be any source of 3D models, e.g., 3D model web store, or other web based source of 3D models, such like Thingiverse, Shapeways, Cubify, GrabCad, Amazon, eBay, etc. Marketplace is a Front end solution that connects to the Master Server through the front end API F. For an end customer it is possible to initialize secured streaming of a 3D model from marketplace to a 3D printer of his choice, paying printing licence fee, choosing parameters for printing, initialize streaming of the model partially or at once to the 3D printer via a secured protocol. Moreover it is possible to distribute secured 3D models via email, facebook, twitter etc. This will lead to a web page (marketplace) with the possibility to buy and start streaming.

[0090] Back end is a system for management of 3D files by a right holder. Right holder can upload and protect 3D files, choose where they would like to publish these files for sales (e.g., on which Marketplaces), to assign descriptions, tags and keywords to files, choose number of prints allowed, set a price for every print, see a distribution statistics of 3d files, or to unpublish files from stores,

[0091] 3D printers could be registered with the Master Server at the stage of manufacturing or during usage.

[0092] The Secured Storage resides on an encrypted segment of storage. This encrypted storage segment could be decrypted only by several human beings or any automation tool outside of the Master Server, so that if the server is physically stolen the database with 3D objects is not recoverable by a third party.

[0093] One example of the method according to present invention is depicted on FIG. **2**. The method comprises the steps of receiving a request to print a 3D object (3D Model, 3D printer) **1000**, checking permissions to print the 3D object at Master Server **1001**, Creating a Virtual Machine for printing said 3D object **1002**, said Virtual Machine checking in at said Master Server **1003**, Authenticating said 3D printer at said Virtual Machine **1004**, said Virtual Machine retrieving a 3D model from a Secured Storage **1005**, said Virtual Machine calculating and streaming instructions for 3D printer **1006**, said Virtual Machine Monitoring the printing progress **1007**, Destroying the Virtual Machine when printing is completed **1008**.

1. A streaming method in a secure manufacturing system comprising a streaming server and a numerically controlled manufacturing machine connected to said streaming server over a communication channel, the method comprises providing to the streaming server a 3D model of a 3D object to be manufactured by said manufacturing machine characterized in that the method additionally comprises on said streaming server, converting said 3D model into a set of manufacturing machine specific instructions for operating said manufacturing machine; encoding said set of instructions into a set of encoded instructions by applying simultaneously or in sequence at least one of the processes selected from the group consisting of calculating a set of hashed instructions by applying a cryptographic hash function to said set of instructions, calculating a set of obfuscated instructions by applying obfuscation function to said set of instructions, applying arithmetic coding to said set of instructions, applying digital fingerprints, calculating checksums, calculating hash values, calculating digital DNA, and encrypting said set of instructions; and

outputting said set of instructions to said manufacturing machine over said communication channel.

2. A method as in claim **1**, comprising providing a server hash table on said streaming server; hashing said set of instructions into a hashed set of instructions, using said server hash table; and outputting said hashed set of instructions as a hashed stream of instructions to said manufacturing machine over said communication channel.

3. A method as in claim **2**, comprising on the manufacturing machine receiving said hashed stream; calculating on said manufacturing machine a local hash table, corresponding to said server hash table; converting said hashed stream, using said local hash table into a stream of instructions and outputting said converted stream of instructions to operate the operational part of the manufacturing machine.

4. A method as in claim **2**, comprising during said hashing repeatedly regenerating said hash table and correspondingly regenerating said local hash table during said converting said hashed stream according to a predetermined algorithm.

5. A method as in claim **1**, comprising splitting said set of instructions into split sets of instructions, obfuscating each of said split sets of instructions, hashing each of said obfuscated splits, streaming said hashed obfuscated splits independently over said communication channel from the streaming server to the manufacturing machine, converting said streamed splits into split sets of instructions and combining said split sets of instructions into the stream of instructions for controlling the manufacturing machine.

6. A method as in claim **1**, wherein said providing said 3D model comprises creating a secure connection over a communication channel between the streaming server and a source of 3D models, hashing said 3D model at the source of 3D models, transferring said hashed 3D model to said streaming server, and re-hashing said hashed 3D model for streaming to said manufacturing machine.

7. A method as in claim **1**, creating a virtual machine on said streaming server for each instance of streaming said 3D model and destroying said virtual machine after said instance of streaming said 3D model is completed.

8. A method as in claim **7**, comprising destroying said virtual machine and creating new virtual machine instance so that each instance of streaming is carried out by more than one virtual machine.

9. A method as in claim **7**, comprising creating more than one virtual machine for each instance of streaming, so that different parts of said 3D model are streamed by different virtual machines.

10. A method as in claim **1**, wherein the system comprises a computer device, comprising a source of 3D models, said computer device connected to said streaming server over a communication channel, the method comprising creating on said computer device a first virtual machine for providing said 3D model to said streaming server, hashing said 3D model in said first virtual machine, creating a secured virtual machine instance on said streaming server, receiving hashed 3D model by said secured virtual machine instance, storing said hashed 3D model in memory hash table, materializing said secured virtual machine instance into hashed virtual machine instance image, said image is transferred to a second computer device connected to a manufacturing machine, executing said secured virtual machine instance on said second computer device and streaming locally said hashes of the 3D model to said manufacturing machine.

11. A method as in claims **1**, wherein said secure manufacturing system comprising a plurality of streaming servers, each streaming server connected to Internet and said steps of secure streaming are carried out by more than one streaming server in concert.

12. A method as in claim **11**, comprising each of said streaming servers streaming a different part of said 3D model to be manufactured.

13. A secure numerically controlled manufacturing system, comprising a streaming server; comprising a conversion module adapted for receiving a 3D model representing a 3D object to be manufactured and converting said 3D model into a set of manufacturing instructions, an obfuscating and hashing module adapted to obfuscate and to hash said set of manufacturing instructions into a hashed set of instructions, a dynamic hash tables database adapted to provide hash tables for said hashing module and a precise time based pseudo number generator module; a source of 3D models, connected to said streaming server over a communication channel; and a manufacturing machine, connected to said streaming server over a communication channel, said manufacturing machine comprising an operational module, a hash lookup module for converting said hashed set of instructions, a Dynamic Local Hash Tables Database for providing hash tables for hash lookup module and precise time based pseudo number generator module for independently synchronizing the hash tables of the manufacturing machine with the hash tables used on said streaming server.

14. A system as in claim **13**, comprising a plurality of streaming servers, each of said streaming servers connected to Internet and adapted perform said secure streaming in concert.

15. A system for secure 3D printing, comprising a 3D printer, comprising a secured module, and connected to a Cloud over said secure module; a Master Server located in the Cloud, said Master Server comprising a front-end application programming interface for Front End API F and an application programming interface for the back end API B, wherein at least one Marketplace for providing 3D models is connected to the Master Server with through the API F, the system further comprising a Secure Storage for 3D models, wherein said 3D models can be uploaded into a Secure Storage in the Cloud using back end through the API B, wherein the Master Server is adapted to receiving a request to print a 3D object, checking permissions to print the 3D object at Master Server, creating a Virtual Machine for printing said 3D object, said Virtual Machine is adapted for checking in at said Master Server, authenticating said 3D printer at said Virtual Machine, said Virtual Machine adapted for retrieving a 3D model from a Secured Storage, said Virtual Machine adapted for calculating and streaming instructions for 3D printer, said Virtual Machine adapted for monitoring the printing progress, and destroying the Virtual Machine when printing is completed.

16. A method of secure streaming for 3D printing, the method comprises the steps of receiving a request to print a 3D object, checking permissions to print the 3D object at Master Server, creating a Virtual Machine for printing said 3D object, said Virtual Machine checking in at said Master Server, authenticating said 3D printer at said Virtual Machine, said Virtual Machine retrieving a 3D model from a Secured Storage, said Virtual Machine calculating and streaming instructions for 3D printer, said Virtual Machine Monitoring the printing progress, and destroying the Virtual Machine when printing is completed.

* * * * *

Appendix 7 - Corrigendum

Additional comments on paper are below.
In the Publication II with the notation $H^{-1}(h) \in \varnothing$ we mean that $H^{-1}(h)$ does not exist.

Curriculum Vitae

1. Personal data

Name	Anton Vedeshin
Date of birth	20 October 1984
Nationality	Estonian

2. Contact information

Address	Tallinn University of Technology, School of IT, Department of Computer Science, Ehitajate tee 5, 19086 Tallinn, Estonia
E-mail	anton.vedeshin@taltech.ee

3. Education

2010–2020	Tallinn University of Technology, School of IT, Computer Science, PhD studies
2006–2009	Tallinn University of Technology, Faculty of Computer Science, Business Information Technology, MSc
2003–2006	Tallinn University of Technology, Faculty of Computer Science, Business Information Technology, BSc
2000–2003	Science School of Tartu University, Programming, Math, Physics School diploma
1996–2003	Mustamäe Science Gymnasium of Tallinn, High School Gymnasium diploma

4. Language competence

Russian	native
Estonian	fluent
English	fluent
German	beginner

5. Professional experience

2013– ...	3D Control Systems, Inc., Founder, CTO
2013– ...	TalTech, Visiting lecturer, Cloud Computing course
2005–2013	Innovative Technologies and Business Systems, Founder, CTO

6. Computer skills

- Programming languages: C, C++, GoLang, Python, Java, Haskel, PHP.
- Databases: Cassandra, Aerospike, Redis, Postgres, MSSQL.
- Frameworks: Hadoop, Spark.
- Blockchains: Ethereum, Corda R3, Hyperledger.
- Cloud platforms: AWS, Azure, GCE, GovCloud.

- Data mining packages: KNIME, Orange, RapidMiner, Radoop, H2O.
- Electronics: Arduino, Raspberry Pi, I2C, ISP, PWM.
- Operating systems: Debian Linux, Kali Linux, BitKey Linux, macOS, Windows.
- CAD: Inventor, SolidEdge, SolidWorks, Fusion 360
- Document preparation: Overleaf, LaTex.

7. Honours and awards

- 2020, Award for the best technical scientific paper of the year 2019, TalTech
- 2016, Leaderboard of Disrupt SF 2016 Battlefield, TechCrunch Disrupt San Francisco
- 2014, Winner of the Positive Change award at the 2014 3D Printshow Global Awards
- 2013, 1st Prize in Business Model competition Mektory, Tallinn University of Technology
- 2012, 1st Price in European Innovation Academy business model competition, Tallinn
- 2012, Finalist of Brainhunt competition (Ajujaht), Tallinn
- 2009, 1st Prize in UNICA Business plan competition in Estonian round, Network of Universities from the Capitals of Europe
- 2004, 2nd Prize in Multimedia competition Tallinn University of Technology
- 2003, Silver Medal Best School Leaver award by Estonian president Arnold Rüütel
- 2003, 2nd Prize in Software Programming Competition, Tallinn University of Technology
- 2001, 8th Prize in Estonian Programming Olympiad, Tartu University
- 2001, 2nd Prize in Tallinn's Programming Olympiad, Tallinn
- 2000, 1st Price in the multimedia competition "Health to children – NO to drugs", Tallinn

8. Defended theses

- 2009, Conformity Calculation using Hadoop Map/Reduce, MSc, supervisor Prof. Innar Liiv, Tallinn University of Technology, Institute of Information Technology
- 2006, Intelligent Information Retrieval from Web Pages, supervisor Dr. Innar Liiv, Tallinn University of Technology, Institute of Information Technology

9. Field of research

- Cyber Security, Automated Manufacturing, Robotics, 3D Printing, Data Mining

10. Scientific work

Papers

1. A. Vedeshin, J. M. U. Dogru, I. Liiv, D. Draheim, and S. Ben Yahia. A digital ecosystem for personal manufacturing: An architecture for cloud-based distributed manufacturing operating systems. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, MEDES '19, page 224–228, New York, NY, USA, 2019. Association for Computing Machinery

2. A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. *IEEE Access*, 2019

3. P.-M. Sepp, A. Vedeshin, and P. Dutt. Intellectual property protection of 3d printing using secured streaming. In *The Future of Law and eTechnologies*, pages 81–109. Springer, 2016

4. A. Vedeshin, J. M. U. Dogru, I. Liiv, S. B. Yahia, and D. Draheim. Smart cyber-physical system for pattern recognition of illegal 3d designs in 3d printing. In *Communications in Computer and Information Science*, pages 74–85. Springer International Publishing, 2020

5. A. Norta, A. Vedeshin, H. Rand, S. Tobies, A. Rull, M. Poola, and T. Rull. Self-aware agent-supported contract management on blockchains for legal accountability

6. A. Vedeshin. Advanced information retrieval from web pages. In *BCS IRSG Symposium: Future Directions in Information Access 2007*, pages 1–6, 2007

7. I. Liiv, A. Vedeshin, and E. Täks. Visualization and structure analysis of legislative acts: a case study on the law of obligations. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 189–190, 2007

Patent Applications

1. K. Isbjornssund and A. Vedeshin. Method and system for enforcing 3d restricted rights in a rapid manufacturing and prototyping environment, Feb. 27 2014. US Patent App. 13/973,816

2. K. Isbjörnssund and A. Vedeshin. Secure streaming method in a numerically controlled manufacturing system, and a secure numerically controlled manufacturing system, Dec. 3 2015. US Patent App. 14/761,588

3. J. K. Isbjörnssund and A. Vedeshin. Optimized virtual 3d printing build tray allocation, Feb. 19 2015. WO Patent App. WO2015022572A2

Elulookirjeldus

1. Isikuandmed

Nimi	Anton Vedešin
Sünniaeg	20.10.1984
Kodakondsus	Eesti

2. Kontaktandmed

Aadress	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduste Instituut, Ehitajate tee 5, 19086 Tallinn, Estonia
Telefon	+372 5562 0101
E-post	anton.vedeshin@taltech.ee

3. Haridus

2010–2020	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Info- ja kommunikatsioonitehnoloogia, doktoriõpe
2006–2009	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Äriinfotehnoloogia, MSc
2003–2006	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Äriinfotehnoloogia, BSc
2000–2003	Tartu Ülikooli Täppisteaduste Kool, Informaatika, Matemaatika, Füüsika diplom
1996–2003	Tallinna Mustamäe Reaalgümnaasium diplom, hõbemedal, EV presidendi vastuvõtt: parim koolilõpetaja 2003

4. Keelteoskus

vene keel	emakeel
eesti keel	kõrgtase
inglise keel	kõrgtase
saksa keel	algtase

5. Teenistuskäik

2013– …	3D Control Systems, Inc., kaasasutaja, tehniline direktor
2013– …	TalTech, külalislektor, Pilvetehnoloogia kursus
2005–2013	Innovative Technologies and Business Systems, kaasasutaja, tehniline direktor

6. Arvutioskus

- Programmeerimiskeeled: C, C++, GoLang, Python, Java, Haskel, PHP.
- Andmebaasid: Cassandra, Aerospike, Redis, Postgres, MSSQL.
- Raamistikud: Hadoop, Spark.
- Plokiahelad: Ethereum, Corda R3, Hyperledger.
- Pilveplatvormid: AWS, Azure, GCE, GovCloud.

- Teadustarkvara paketid: KNIME, Orange, RapidMiner, Radoop, H2O.
- Elektroonika: Arduino, Raspberry Pi, I2C, ISP, PWM.
- Operatsioonisüsteemid: Debian Linux, Kali Linux, BitKey Linux, macOS, Windows.
- CAD: Inventor, SolidEdge, SolidWorks, Fusion 360
- Dokumentide ettevalmistus: Overleaf, LaTex.

7. Autasud

Autasude loetelu on toodud ingliskeelse elulookirjelduse juures.

8. Kaitstud lõputööd

- 2009, “Konformismi kalkuleerimine kasutades Hadoop Map/Reduce”, MSc, juhendaja Prof. Innar Liiv, Tallinna Tehnikaülikool, Tarkvarateaduste Instituut
- 2006, “Intelligentne informatsiooni hankimine veebilehtedelt”, BSc, juhendaja Dr. Innar Liiv, Tallinna Tehnikaülikool, Tarkvarateaduste Instituut

9. Teadustöö põhisuunad

- Küberturvalisus, automatiseeritud tootmine, robootika, 3D-printimine, andmekaevandamine

10. Teadustegevus

Patentide, teadusartiklite, konverentsiteeside ja konverentsiettekannete loetelu on toodud ingliskeelse elulookirjelduse juures.