e-ISSN: 0974-4614
p-ISSN: 0972-0448

Invisi Guard: Blockchain-Powered Invisible Code Technology for Anti-Counterfeit Assurance

Jeenath Laila N[1*], Mohamed Rizwan[2]

[1]Assistant Professor, Government College of Engineering, Tirunelveli-7
[2]Research Scholar, Karunya Institute of Technology and Sciences, Coimbatore
Email: jeenathalila@gcetly.ac.in
*Corresponding Author

Received: 17.08.2024 Revised: 10.09.2024 Accepted: 09.10.2024

ABSTRACT
Counterfeit products are a major concern for businesses and consumers alike, as they can lead to significant financial losses and damage to brand reputation. To combat this problem, a blockchain-based system using Invisible Quick Response(QR) codes has been proposed.This paper focuses on a novel approach for storing invisible QR code images using the InterPlanetary File System (IPFS) and recording the Content Identifier (CID) in a blockchain system to enhance counterfeit avoidance in e-commerce websites. Traditional QR codes are visible and can be easily replicated, making them susceptible to counterfeiting. By utilizing invisible QR codes, the proposed method adds an extra layer of security. The IPFS is employed as a decentralized storage system, allowing the QR code image to be securely stored and accessed. The CID, a unique identifier for the stored image, is then recorded in a blockchain, providing an immutable and transparent record of the QR code's authenticity. Additionally, a name server and Access Control List (ACL) are implemented to facilitate efficient retrieval of QR code information while maintaining security and privacy. This approach enhances trust and authenticity in e-commerce transactions by enabling buyers to validate the legitimacy of products through the blockchain-recorded CID associated with the invisible QR code. The proposed system provides an effective solution to mitigate counterfeiting and build consumer confidence in online purchase.

Keywords: Counterfeit ; Blockchain; IPFS; AES encryption; access control; name server; Proof of Fair Chance consensus mechanism; invisible QR code

1. INTRODUCTION

E-commerce, short for electronic commerce, refers to the buying and selling of goods and services over the internet. It involves conducting business transactions electronically, eliminating the need for physical stores or face-to-face interactions. E-commerce has experienced tremendous growth and has become an integral part of the global economy.The rise of the internet and advancements in technology have revolutionized the way businesses operate, enabling them to reach customers worldwide and conduct transactions online. E-commerce offers numerous advantages for both businesses and consumers, including convenience, accessibility, a wide range of product choices, competitive pricing, and personalized shopping experiences. Counterfeiting refers to the production, distribution, or sale of imitation or fake products that are designed to mimic the appearance and branding of genuine products. Counterfeit products are a significant concern in the e-commerce industry, posing risks to both consumers and legitimate businesses. A counterfeit product is, by definition, a poor imitation of an authentic good. Its major goal is to imitate an expensive product while charging less so that customers can save money. These days, these products' quality is essentially comparable to that of the original. The Organisation for Economic Co-operation and Development (OECD) reports that during the past few years, the global trade in counterfeit goods has grown steadily and currently accounts for 3.3% of all trade. Depending on the type of items, such as food, medication, and cosmetic products, counterfeiting can potentially impair consumers' health by stealing sales from legitimate businesses. Many online businesses are investing in strategies to lower the amount of fake goods discovered on their websites. For instance, since launching Project Zero, Amazon has reportedly prevented ten billion attempts to list counterfeit goods and destroyed two million of them that were already in its storage. A machine-learning technology called Project Zero eliminates products that have been recognised as possibly being phony. To guard against fraud, Amazon has hired 10,000 people and spent more than $700 million.These problems have increased the need for a trustworthy method of product authentication prior to purchase, especially if the products are being traded between customers. Today, blockchain technology offers a dependable approach to building a foundation of consumer trust. Sales of counterfeit goods will be drastically decreased while allowing customers to shop with confidence by developing a distributed ledger with consensus, provenance, immutability, and finality to enable consumers to ensure the authenticity of their products.

To store and retrieve data on the blockchain, various protocols and technologies are used, including InterPlanetary File System (IPFS). IPFS is a peer-to-peer protocol that allows for the storage and retrieval of files in a decentralized manner. It provides a distributed and resilient file system that is not dependent on a central server, making it ideal for use in blockchain-based systems. With IPFS, data is stored across multiple nodes in the network, making it difficult for any single node to modify or delete the data.

To ensure the security and privacy of data stored on the blockchain, encryption techniques such as Advanced Encryption Standard (AES) are used. AES is a widely adopted encryption standard that provides strong encryption and decryption capabilities, making it ideal for securing data in blockchain-based systems. With AES encryption, data is encrypted using a secret key, making it difficult for unauthorized users to access the data.

Smart contracts are another key feature of blockchain technology that allows for the execution of automated and self-executing contracts. Smart contracts are self-executing code that automatically execute the terms of a contract when certain conditions are met. They enable the creation of trustless and decentralized applications, making blockchain-based systems more efficient and transparent.

Access control is another important aspect of blockchain-based systems. Access control mechanisms ensure that only authorized users can access and modify data on the blockchain. This is achieved through the use of public and private keys, which are used to authenticate and authorize access to data stored on the blockchain. With access control mechanisms, users can have greater control over their data and prevent unauthorized access to sensitive information.

Finally, name servers play a crucial role in blockchain-based systems by mapping human-readable domain names to IP addresses. Name servers enable users to access resources on the blockchain using human-readable domain names, making the system more user-friendly and accessible. With name servers, users can access resources on the blockchain without having to remember complicated IP addresses.

The use of blockchain technology can help to increase transparency, reduce fraud, and improve efficiency in these industries.However, there are also challenges associated with the use of blockchain technology, including scalability, interoperability, and regulatory compliance. As blockchain-based systems become more widespread, it is important to address these challenges and develop solutions that can enable the technology to reach its full potential.

An invisible QR code, also known as a transparent QR code, is a type of QR code that is designed to blend seamlessly into its surroundings. Unlike traditional QR codes, which are typically black squares on a white background, an invisible QR code is designed to be transparent or nearly transparent.The purpose of an invisible QR code is to make the code itself less noticeable, allowing it to be placed on objects or surfaces without detracting from their appearance. This can be useful in situations where you want to provide QR code functionality without visually disrupting the design or aesthetics of the object or surface.To create an invisible QR code, special techniques and materials are used to generate a QR code that can be applied to a surface while maintaining its transparency. These techniques often involve using high-contrast materials or using UV or infrared inks that are invisible to the naked eye but can be detected by QR code scanners. When scanning an invisible QR code, a QR code scanner app or device will still be able to recognize and decode the code, even though it may not be visible to the human eye. This allows users to access the encoded information, such as a website URL, text, or other data, by scanning the code with a compatible device.

The contribution of the paper is as follows:

- We propose a model with invisible code to detect the authenticated and fake product.
- We propose a blockchain data storage and retrieval model that illustrates how to use IPFS networks to reduce the storage of blockchain data and also supports security of data.
- We implement a Proof of Fair chance showing how miners can use this model in this paper to store less blockchain data in real mining scenarios, and how new nodes can quickly synchronize with the network.

The organization of the paper is as follows. Section 2 describes the review of research work in the areas of blockchain data storage and explains the different techniques implemented for traditional data storage. Section 3 describes the privacy-preserving storage and retrieval model in detail, including system design and data processing flow for storage and retrieval. Section 4 calculates the size and time for data storage and retrieval and analyzes the performance of the model. Section 5 is a summary and future work.

2. LITERATURE REVIEW

Jin sun et.al. proposes a secure storage and access scheme for Electronic Medical Records (EMRs) using blockchain and IPFS. The scheme addresses the security, privacy, and accessibility issues of EMRs by leveraging the decentralized and immutable nature of blockchain and IPFS. The EMRs are encrypted and stored on IPFS, while the metadata and access control information are stored on the blockchain. The scheme ensures transparency and auditability of changes made to the EMRs.The scheme provides enhanced security and privacy for EMRs through encryption and blockchain-based access control. It also improves accessibility and transparency by leveraging IPFS and blockchain technology. The complexity of the scheme may require specialized knowledge to implement and maintain, and the scalability of the system may be limited due to the constraints of blockchain and IPFS technology. Additionally, the system may face interoperability issues with existing healthcare systems, necessitating significant changes to integrate with them.[1]

M. Ali et al. presents a decentralized alternative to traditional internet infrastructure, which uses a combination of blockchain technology, peer-to-peer networking, and cryptographic algorithms to provide security and reliability. The advantages of Blockstack include its decentralized nature, which makes it resistant to censorship and provides users with control over their data. It also uses a unique naming system that allows users to create their own identities and associate them with their digital assets. One of the main challenges is user adoption, as the system requires users to install software and manage their own private keys. This can be difficult for non-technical users, which limits the potential user base. Additionally, the reliance on blockchain technology can make the system slow and expensive to use, particularly as the number of users and transactions increases. [2].

V. Rathore et al. proposes a secure storage and retrieval system for encrypted data in IPFS using Shamir's Secret Sharing scheme. The system is designed to provide secure storage and retrieval of data in IPFS, a peer-to-peer network for storing and sharing files. The system encrypts the data using AES and then divides it into multiple shares using Shamir's Secret Sharing scheme. These shares are then distributed across the network, making it difficult for an attacker to retrieve the original data without possessing a sufficient number of shares.The advantages of this system include its high level of security, as the combination of AES encryption and Shamir's Secret Sharing provides strong protection against attacks. One of the main challenges is the complexity of managing the shares, which requires careful coordination between the various nodes in the network. Additionally, the system may suffer from reduced performance and increased latency due to the need to distribute and retrieve multiple shares of the data [3]

M. Afzal et al. proposes a system for secure data sharing in IPFS using blockchain-based identity management. The system uses a combination of IPFS and blockchain technology to provide secure and decentralized storage and sharing of data. This ensures that only authorized users can access the shared data, and that their actions are recorded on the blockchain for accountability and auditability.The advantages of this system include its high level of security, as the combination of AES encryption, Shamir's Secret Sharing, and blockchain-based identity management provides strong protection against attacks.The system may require significant computational resources, particularly for the encryption and decryption of large volumes of data.[4]

Blockchain provides a platform for decentralization and trust in various applications such as finance, commerce, IoT, reputation systems, and healthcare. However, prevailing challenges like scalability, resilience, security and privacy are yet to be overcome. Due to rigorous regulatory constraints such as HIPAA, blockchain applications in the healthcare industry usually require more stringent authentication, interoperability, and record sharing requirements. Healthcare data management faces challenges like transparency, security, and centralization risks. Blockchain offers a decentralized solution to enhance trust, privacy, and resilience in healthcare systems. This paper explores blockchain’s potential, key features, and real-world applications in healthcare. It also identifies challenges and future research directions for successful adoption [5].

Blockchain is getting large adoptions for various applications besides cryptocurrencies. Despite these benefits, scalability is a big challenge to blockchain impeding its mainstream adoption. This paper[6] gives a systematic review of blockchain scalability. It follows a systematic process to investigate the research trend on blockchain scalability and review its state of the art. It reviews the various proposed solutions and methods for blockchain scalability. It also reviews the performance analysis of blockchain systems. This paper assesses the proposed scalability solutions, deduce future research directions on the blockchain scalability, and finally discuss the blockchain adoption.

Masaki Fujikawaa and et al [7] presented a novel method for embedding nearly invisible 2D codes containing both public and secret information onto ceramic products. The goal is to provide a discreet and secure way of labeling ceramic items while enabling access to relevant information using a QR code scanner.The method involves the utilization of a specialized ink formulation that is transparent to the naked eye but visible under specific lighting conditions or using specialized scanners. This ink is applied to the ceramic surface before the firing process, ensuring that the code becomes an integral part of the product.The nearly invisible nature of the 2D code allows for discreet labeling of ceramic products without visually detracting from their aesthetic appeal.The method enables the inclusion of both public and secret information within the 2D code. Public information can be accessed by anyone scanning the code, while secret information requires additional authentication or specialized scanning equipment.By embedding secret information within the code, the method offers an added layer of security, making it difficult for unauthorized individuals to access sensitive data.The firing process ensures that the 2D code becomes a permanent part of the ceramic product, making it resistant to wear, tear, and other environmental factors.The transparency of the code may result in decreased readability, especially under poor lighting or with low-quality scanners. Testing and optimization are necessary to ensure consistent and reliable code scanning.The specialized ink and equipment required for this method may involve additional costs for manufacturers. Moreover, implementing the necessary scanning infrastructure could require adjustments to existing processes and workflows.

Zhong Pai Gao and et al proposed a new method to embed QR Code on a digital screen via temporal psychovisual modulation (TPVM)[8]. By exploiting the difference between human eyes and semiconductor imaging sensors in temporal convolution of optical signals, we make QR Code perceptually transparent to humans but detectable for mobile devices. Based on the idea of invisible QR Code, many applications can be implemented, e.g., “physical hyperlink” for something interesting on TV or digital signage , “invisible watermark” for anti-piracy in theater. A prototype system introduced in this paper serves as a proof-of-concept of the invisible QR Code and can be improved in future works.

With the development of commodity economy, the emergence of fake and shoddy products have seriously harmed the interests of consumers and enterprises. To tackle this challenge, Rongjun Chen and et al proposed customized 2D barcodes to satisfy the requirements of the enterprise anti-counterfeiting certification. Based on information hiding technology, [9] the proposed approach can solve these challenging problems and provide a low-cost, difficult to forge, and easy to identify solution,while achieving the function of conventional 2D barcodes. By weighting between the perceptual quality and decoding robustness in sensing recognition, the customized 2D barcode can maintain a better aesthetic appearance for anti-counterfeiting and achieve fast encoding.

Ren and et al [10] proposed a method, DCOMB (dual combination Bloom filter), combining the data stream of the IoT (Internet of Things) with the timestamp of the blockchain,to improve the versatility of the IoT database system.

N. R. Pradhan et al. in [11] A Google Cloud Platform-based multi-organizational, multi-host, off-chain and on-chain framework for keeping track of patient medical information as well as various peer-based plans for a hyperledger fabric-enabled medical system that addresses the issues of data privacy, data availability, and data security have been proposed. They used tcpdump to generate realistic network traffic for their performance analysis, orderer for RAFT, and Kafka for their performance research to examine the system's performance. Furthermore, they contrasted the orderer services offered by Kafka and RAFT, finding that RAFT was more appropriate for open, query, and client-side transfer operations.

3. MATERIALS AND METHODS

The proposed system aims to create a comprehensive solution for product authentication and verification. It involves a series of modules that collectively ensure the integrity and traceability of products throughout their lifecycle. The system begins with the registration of manufacturers, followed by their authentication to establish trust and credibility. Product registration is then performed, wherein invisible 2D barcodes are generated using a certified algorithm, incorporating manufacturer certification details. The generated barcodes are printed on product covers, enabling subsequent scanning. The system leverages IPFS storage and blockchain technology to securely store transaction data, ensuring transparency and immutability. A proof-of-fair-chance consensus mechanism is employed to maintain the integrity of the system. When needed, product information can be retrieved from the blockchain for verification purposes. Ultimately, the system facilitates product verification, enabling consumers and stakeholders to validate the authenticity and origin of products, fostering trust and combating counterfeiting. The proposed system architecture was depicted in the figure 1

3.1. Manufacturer Registration

Manufacturer registration is the process of adding a new manufacturer to the system. When registering as a manufacturer on an e-commerce platform, the following details that are typically requested during manufacturer registration:

- Business Name: The legal or registered name of a manufacturing business.
- Contact Information: This includes their business address, email address, and phone number. It helps customers and the e-commerce platform to reach out to the manufacturer for inquiries and communication.
- Business Description: A brief description of their manufacturing business, including the types of products produced and any unique selling points.
- Business Logo: The company's logo is requested for branding purposes on the e-commerce platform.
- Product Catalog: It contains the details about the products manufactured, such as product names, descriptions, specifications, pricing, and images. Product codes or SKUs (Stock Keeping Units) should be provided for inventory management.
- Manufacturing Certifications: Manufacturing certifications or quality assurance standards for the manufacturing product.
- Tax Information: Tax Identification Number (TIN) or Goods and Services Tax (GST) number.
- Shipping and Returns Policy: Information about the manufacturers shipping methods, estimated delivery times, and return/refund policies. This helps customers understand the terms and conditions associated with purchasing your products.

- Payment Details: Payment details, such as bank account information or preferred payment gateway integration, to receive payments for your products sold on the e-commerce platform.
- Legal Documentation: Legal documents, such as business licenses, permits, or registrations, to validate the status of a manufacturer.

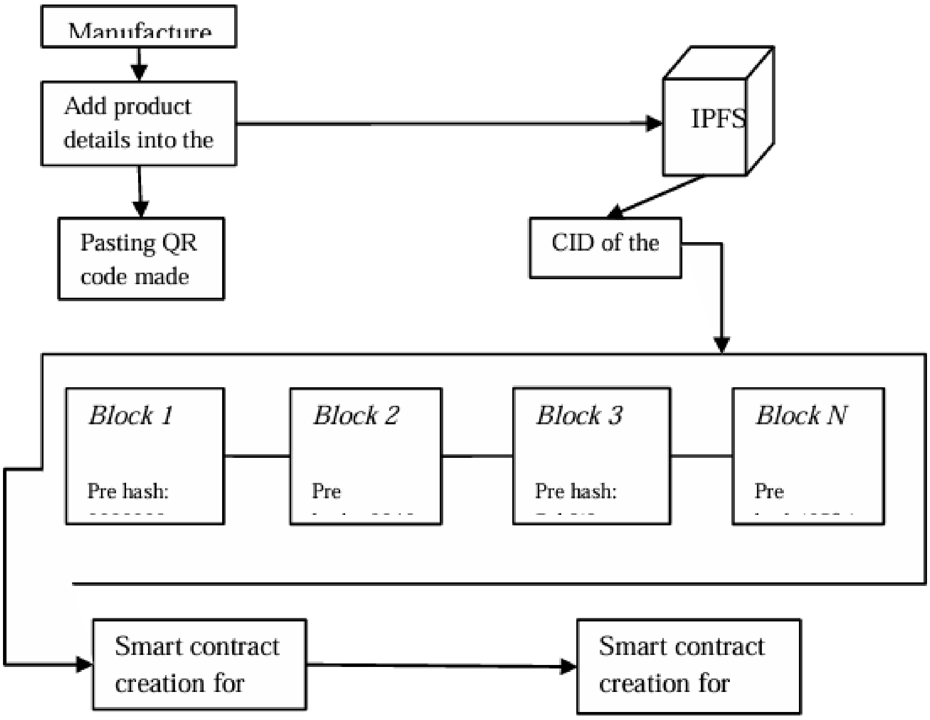
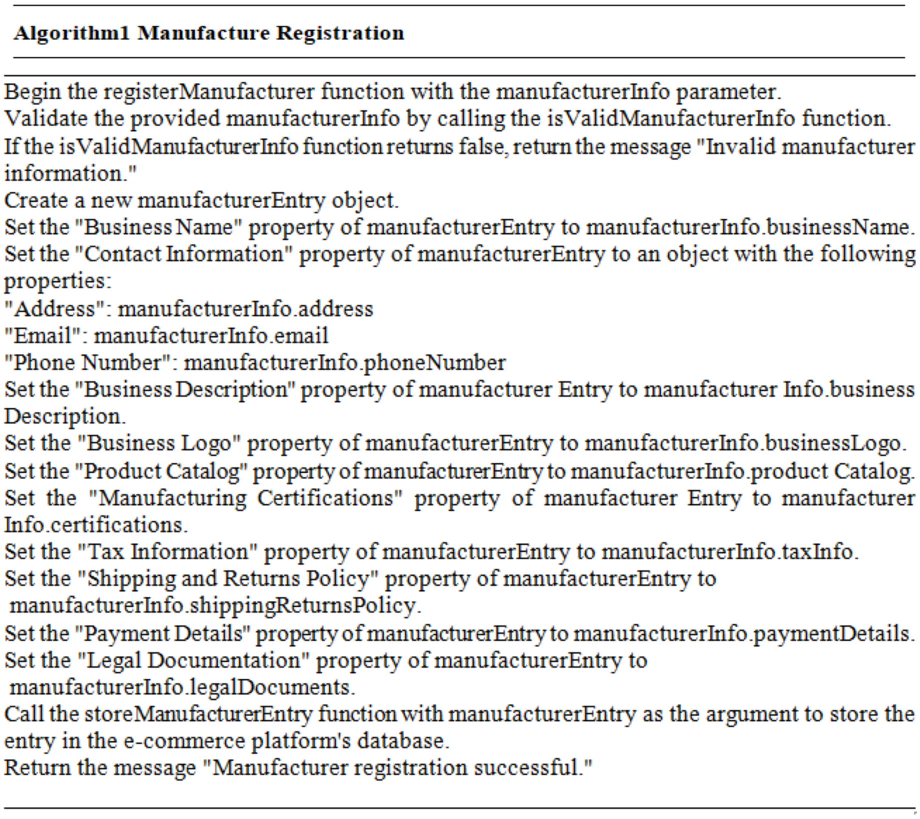


Figure 1. Proposed System for Storing the product details



Algorithm1 Manufacture Registration

```
Begin the registerManufacturer function with the manufacturerInfo parameter.
Validate the provided manufacturerInfo by calling the isValidManufacturerInfo function.
If the isValidManufacturerInfo function returns false, return the message "Invalid manufacturer information."
Create a new manufacturerEntry object.
Set the "Business Name" property of manufacturerEntry to manufacturerInfo.businessName.
Set the "Contact Information" property of manufacturerEntry to an object with the following properties:
"Address": manufacturerInfo.address
"Email": manufacturerInfo.email
"Phone Number": manufacturerInfo.phoneNumber
Set the "Business Description" property of manufacturer Entry to manufacturer Info.business Description.
Set the "Business Logo" property of manufacturerEntry to manufacturerInfo.businessLogo.
Set the "Product Catalog" property of manufacturerEntry to manufacturerInfo.product Catalog.
Set the "Manufacturing Certifications" property of manufacturer Entry to manufacturer Info.certifications.
Set the "Tax Information" property of manufacturerEntry to manufacturerInfo.taxInfo.
Set the "Shipping and Returns Policy" property of manufacturerEntry to manufacturerInfo.shippingReturnsPolicy.
Set the "Payment Details" property of manufacturerEntry to manufacturerInfo.paymentDetails.
Set the "Legal Documentation" property of manufacturerEntry to manufacturerInfo.legalDocuments.
Call the storeManufacturerEntry function with manufacturerEntry as the argument to store the entry in the e-commerce platform's database.
Return the message "Manufacturer registration successful."
```

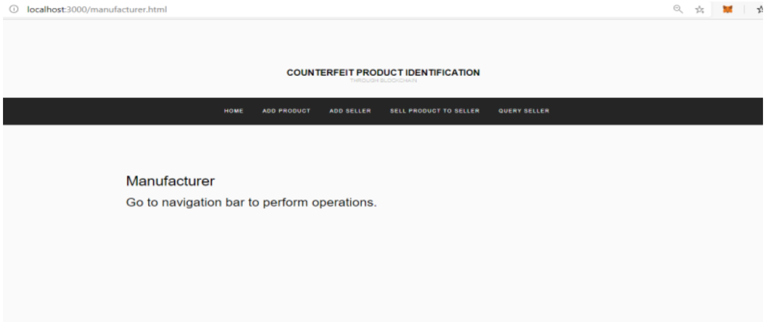
3.2. Manufacturer Authentication

Manufacturer authentication in an e-commerce platform typically involves a verification process to ensure that the manufacturer's identity and legitimacy are confirmed. The following steps are involved in manufacturer authentication:

- Registration: The manufacturer completes the registration process on the e-commerce platform, providing all the necessary details as mentioned earlier.
- Documentation: The manufacturer may be required to submit relevant documents to support their authentication. These documents may include business registration certificates, manufacturing licenses, tax identification documents, and any other certifications specific to the industry.
- Review and Verification: The e-commerce platform reviews the provided information and documents to authenticate the manufacturer's identity and legitimacy. This process may involve manual verification by the platform's staff or an automated verification system.
- Site Visit or Inspection (if applicable): In some cases, the e-commerce platform may conduct a site visit or inspection of the manufacturer's facilities to validate their manufacturing capabilities, quality control processes, and adherence to industry standards.
- Brand or Trademark Verification: If the manufacturer claims to have registered trademarks or brand names, the e-commerce platform may verify the authenticity of those claims by checking the relevant trademark databases or requesting supporting documents.
- Communication and Clarification: During the authentication process, the e-commerce platform may communicate with the manufacturer to seek additional information or clarification regarding their business operations, products, or any other relevant details.
- Approval or Rejection: Based on the review, verification, and authentication process, the e-commerce platform determines whether to approve or reject the manufacturer's authentication request. The manufacturer is then notified of the outcome.

Algorithm 2 Manufacturer Authentication

```
Step 1: Declare a function named registerManufacturer with two string parameters _name and _location. It should be a public function.
  1.1. Check if the _name parameter is not empty using the require statement.
  1.2. Check if the _location parameter is not empty using the require statement.
  1.3. Create a new Manufacturer object with the given _name and _location, and set the isVerified flag to false.
  1.4. Store the newly created Manufacturer object in the manufacturers mapping, with the sender's address as the key.
  1.5. Emit an event to indicate the successful registration of the manufacturer.
Step 2: Declare a function named verifyManufacturer with an address parameter _manufacturerAddress. It should be a public function and can only be called by a verified manufacturer.
  2.1. Check if the manufacturer at _manufacturerAddress is not already verified using the require statement.
  2.2. Set the isVerified flag of the manufacturer at _manufacturerAddress to true.
 2.3. Emit an event to indicate the successful verification of the manufacturer.
Step 3: Declare a function named crossReferenceExternalDatabases with a Manufacturer parameter _manufacturer. It should be an internal pure function and returns a boolean value.
  3.1. Check if the name of the _manufacturer is "ACME Corp" and the location is "123 Main St" using the keccak256 function to compare the hashed values.
  3.2. If the condition is true, return true; otherwise, return false.
Step 4: Declare a function named verifyCertifications with a Manufacturer parameter _manufacturer. It should be an internal pure function and returns a boolean value.
4.1. Return true unconditionally.
Step 5: Declare a function named peerValidation with a Manufacturer parameter _manufacturer. It should be an internal pure function and returns a boolean value.
5.1. Return true unconditionally.
Step 6: Declare a function named achieveConsensus with a Manufacturer parameter _manufacturer. It should be an internal pure function and returns a boolean value.
6.1. Return true unconditionally.
```



localhost:3000/manufacturer.html

COUNTERFEIT PRODUCT IDENTIFICATION
THROUGH BLOCKCHAIN

Manufacturer

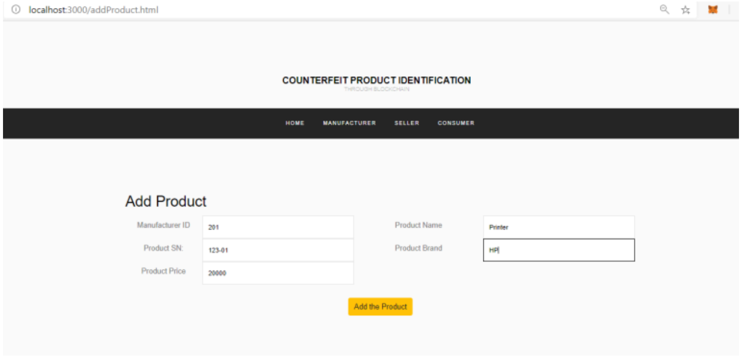
Go to navigation bar to perform operations.

Figure 2 . Manufacturer Page

3.3. Product Registration

Manufacturer registers each and every product manufactured by them with the unique product identifier for each product at the manufacturers site along with the following details:

- Product Name: The name or description of the product.
- Price: The retail or wholesale price of the product.
- Stock Keeping Unit (SKU): A unique identifier used for inventory management and tracking purposes.
- Product Dimensions: The physical dimensions of the product, such as width, height, and depth.
- Manufacturing Date: The date when the product was manufactured or produced.
- Expiration Date: If applicable, the date after which the product is no longer considered safe or effective.
- Batch or Lot Number: A unique identifier assigned to a specific group or batch of products.
- Serial Number: A unique identifier assigned to an individual product unit.
- Country of Origin: The country where the product was manufactured or assembled.
- Manufacturer Information: The name or identification of the manufacturer or brand.
- Regulatory Information: Any regulatory or compliance information associated with the product, such as certifications, safety standards, or FDA (Food and Drug Administration) approvals.



localhost:3000/addProduct.html

COUNTERFEIT PRODUCT IDENTIFICATION
THROUGH BLOCKCHAIN

Add Product

Manufacturer ID: 201
Product SN: 123-01
Product Price: 20000
Product Name: Printer
Product Brand: HP

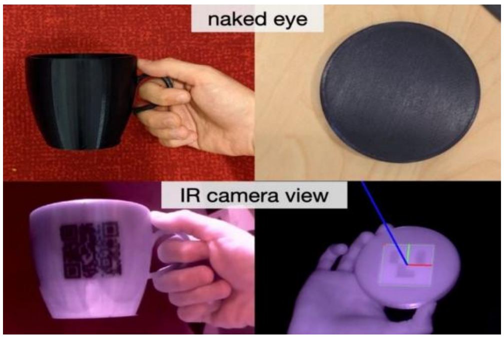
Figure 3. Product Registration page

3. 4. Invisible QR Code Generator

Once the manufacturer has registered each product with its unique product identifier, they can take the process further by embedding these product details into QR codes and printing them using luminescent security ink. This innovative approach combines the power of QR codes with the aesthetic appeal of luminescent security ink, offering a seamless integration of information on the product itself. Using a QR code generator, the manufacturer encodes the product details, including the product name, price, SKU, dimensions, manufacturing date, expiration date, batch or lot number, serial number, country of origin, manufacturer information, and regulatory information. This ensures that a comprehensive set of information is associated with each product. The generated QR code is then printed on the product using luminescent security ink. luminescent security ink allows the QR code to blend seamlessly with the surface, making it nearly invisible to the naked eye. This ensures that the QR code does not compromise the product's visual presentation or branding. The integration of product details in QR codes offers several advantages. Firstly, it optimizes the use of limited space on packaging or labels, as the QR code replaces the need for extensive text or graphics to convey product information. Secondly, it provides consumers with easy access to a wealth of information simply by scanning the QR code with their smartphones. This empowers consumers to make informed purchasing decisions and gain deeper insights into the product's specifications, safety standards, certifications, and manufacturing origin. By utilizing luminescent security ink for printing, manufacturers maintain the product's aesthetic appeal and ensure that the QR code remains inconspicuous. The transparent nature of the ink allows the QR code to seamlessly blend with the product's design, preserving its visual integrity. Furthermore, embedding product details in QR codes enables manufacturers to update information without the need for physical changes to the packaging or labels. This facilitates efficient and timely updates, ensuring that consumers always have access to the most accurate and relevant information about the product.

For example, consider the following product details:

1. Product Name: Widget X
2. Price: $9.99
3. SKU: ABC123
4. Product Dimensions: 5" x 5" x 2"
5. Manufacturing Date: 2023-05-15
6. Expiration Date: 2025-05-15
7. Batch or Lot Number: 4567
8. Serial Number: 123456789
9. Country of Origin: India
10. Manufacturer Information: ABC Manufacturing
11. Regulatory Information: FDA Approved



3.5 Luminescent Security Ink

Luminescent security ink is a specialized type of ink used for anti-counterfeiting purposes. It contains luminescent materials that emit light of specific wavelengths or colors when exposed to certain excitation sources. The synthesis of Ln3+ doped GdPO4 nanorods with tunable crystal structure at different reaction temperatures through the

co-precipitation method. Downshifting and upconversion photoluminescence properties of these nanomaterials are discussed. These nanorods are applied for anti-counterfeiting. Therefore, dual mode emitting luminescent security ink is prepared for making security patterns to combat counterfeiting. The synthesis employs analytical grade chemicals without further purification. Alfa Aesar provided the gadolinium nitrate hydrate (99.9%, Gd(NO3)3.xH2O) and the erbium nitrate pentahydrate (>99.9%, Er(NO3)3.5H2O). Sigma Aldrich sold Europium nitrate pentahydrate (99.9%, Eu (NO3)3.5H2O), Ytterbium nitrate pentahydrate (>99.9%, Yb (NO3)3.5H2O), and S D Fine-Chem Limited sold Ammonium dihydrogen orthophosphate (99.0%, NH4H2PO4). We obtained ethylene glycol (C2H6O2, Emparta grade) from Merck Millipore in India.

Eu3+ doped GdPO4 nanorods are prepared by a simple precipitation method. For this, 627.26 mg of Gd(NO3)3.xH2O and 44.12 mg of Eu(NO3)3.5H2O were dissolved into 20mL ethylene glycol on a hot plate at 60 ◦C. To this solution, 3 mL of the aqueous solution containing 0.3 g of NH4H2PO4 was added under vigorous stirring. Immediately, a white precipitate was formed. Then the temperature was slowly increased to 100/150/185 ◦C, and the resultant solution was stirred for 2 h at that temperature. Then it was cooled down naturally to reach room temperature and washed with an excess amount of methanol and acetone. The final product was kept for drying overnight under ambient conditions. The GdPO4:5% Eu3+ samples prepared at 100, 150, 185 ◦C temperatures were denoted as GDP-100, GDP-150, GDP-185. The GDP-185 sample was calcined at 650 ◦C for 4 h in air and named as GDP-650. The schematic representation of the synthesis methodology of GdPO4:5% Eu3+ samples is presented in Scheme S1 of Supplementary Information.

For the synthesis of GdPO4:1%Er3+, 10 %Yb3+ nanorods, 629.80 mg of Gd(NO3)3.xH2O, 9.14 mg of Er(NO3)3.5H2O, 92.60 mg of Yb(NO3)3.5H2O were dissolved in 20 mL of ethylene glycol. The temperature of the solution was slowely raised to 60 ◦C. To this solution, 3 mL of the aqueous solution containing 0.3 g of NH4H2PO4 was added under vigorous stirring. Then the temperature was raised to 185 ◦C and maintained at this temperature for 2 h followed by cooling to room temperature naturally. The precipitate was separated and washed several times with methanol and acetone. The sample was dried under ambient conditions overnight. 594.42 mg of Gd(NO3)3.xH2O, 9.14 mg of Er(NO3)3.5H2O, 92.60 mg of Yb(NO3)3.5H2O, and 44.12 mg of Eu(NO3)3.5H2O are used as starting materials and the above procedure was followed for the synthesis of GdPO4:1%Er3+, 10 %Yb3+, 5%Eu3+ nanorods.

3.6 Product cover printing

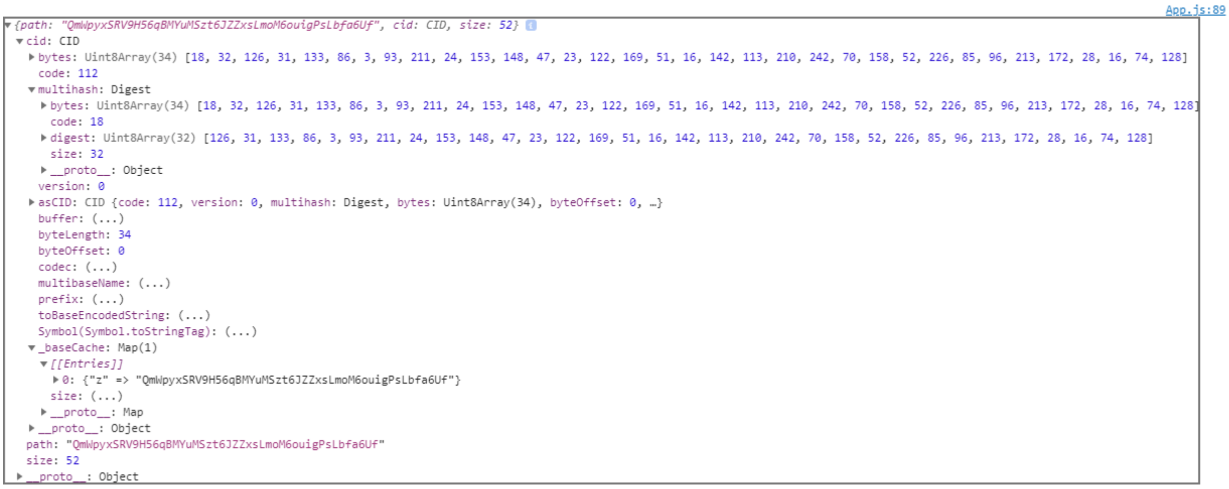
Once the QR codes with embedded product details have been printed using luminescent security ink, the next step is to carefully paste them onto the products. This process ensures that the QR code remains inconspicuous while still accessible to consumers. To begin, manufacturers must determine the optimal placement for the QR code on each product. Factors such as the product's shape, size, and packaging material should be considered to ensure a seamless integration. The QR code should be positioned in a way that does not hinder the product's functionality or compromise its visual appeal. Before applying the QR code, it is crucial to ensure that the product's surface is clean and free from any dust, dirt, or moisture. This ensures proper adhesion of the QR code and prevents any potential issues during scanning.350 mg of the GDP-185 sample, as manufactured, was combined with 5 mL of commercially available polyvinyl chloride (PVC) gold medium ink to create luminous security ink based on GdPO4:Eu3+ nanorods for anti-counterfeiting. To create uniform luminous ink, the mixture was sonicated at 53 KHz for 30 minutes. The ink that was collected was used to create security patterns. Under the printed mesh was a black paper retained for that purpose. After being pressed across the mesh surface, the luminous ink was poured onto the printed mesh. Using 394 nm UV light for downshifting and 980 nm NIR light for upconverting, the security information was read out. The pasting of invisible QR codes on products offers numerous benefits. It provides a discreet and elegant way to integrate product information without detracting from the product's overall aesthetics. Consumers can simply scan the QR code using their smartphones or scanning devices to access a wealth of detailed information about the product, enhancing their overall product experience. Furthermore, the invisible nature of the QR code discourages tampering or removal attempts, as it may go unnoticed by unauthorized individuals. This enhances product security and authenticity, safeguarding against counterfeiting or unauthorized replication.

3. 7. IPFS File Storage

The unique barcode generated for each product is stored in IPFS and the Content Identifier returned by IPFS is stored in blockchain. The steps to store barcode image file in IPFS are as follows:

- Encrypt the file using AES: The bar code image file is encrypted using the AES-128 algorithm.
- Add the encrypted file to IPFS: Use the IPFS command, IPFS add to add the encrypted file to IPFS. This will generate a hash for the file that can be used to retrieve it later. The hash value generated by IPFS is called as Content Identifier(CID)
- Verify the file added to IPFS: Use the IPFS command, IPFS ls to verify that the file was added to IPFS.
- Store the hash in blockchain: The generated hash for the file is then added to the blockchain as a transaction once the file has been spread over the IPFS network in order to maintain the integrity of the file.

CID (hash) of IPFS will be of 256 bits or 32 bytes. The size of the CID of every file or data stored in IPFS will depend on the cryptographic hash algorithm used for hashing. As most content in IPFS is hashed using sha2-256, most CIDs there will be the same size (256 bits, which equates to 32 bytes). The figure 4 shows the file stored in IPFS and obtained the CID



```
                                                                                                                   App.js:89
▼{path: "QmWpyxSRV9H56qBMYuMSzt6JZZxsLmoM6ouigPsLbfa6Uf", cid: CID, size: 52}
 ▼cid: CID
  ▶bytes: Uint8Array(34) [18, 32, 126, 31, 133, 86, 3, 93, 211, 24, 153, 148, 47, 23, 122, 169, 51, 16, 142, 113, 210, 242, 70, 158, 52, 226, 85, 96, 213, 172, 28, 16, 74, 128]
   code: 112
  ▼multihash: Digest
   ▶bytes: Uint8Array(34) [18, 32, 126, 31, 133, 86, 3, 93, 211, 24, 153, 148, 47, 23, 122, 169, 51, 16, 142, 113, 210, 242, 70, 158, 52, 226, 85, 96, 213, 172, 28, 16, 74, 128]
    code: 18
   ▶digest: Uint8Array(32) [126, 31, 133, 86, 3, 93, 211, 24, 153, 148, 47, 23, 122, 169, 51, 16, 142, 113, 210, 242, 70, 158, 52, 226, 85, 96, 213, 172, 28, 16, 74, 128]
    size: 32
   ▶__proto__: Object
   version: 0
  ▶asCID: CID {code: 112, version: 0, multihash: Digest, bytes: Uint8Array(34), byteOffset: 0, …}
   buffer: (...)
   byteLength: 34
   byteOffset: 0
   codec: (...)
   multibaseName: (...)
   prefix: (...)
   toBaseEncodedString: (...)
   Symbol(Symbol.toStringTag): (...)
  ▼_baseCache: Map(1)
   ▼[[Entries]]
    ▶0: {"z" => "QmWpyxSRV9H56qBMYuMSzt6JZZxsLmoM6ouigPsLbfa6Uf"}
    size: (...)
   ▶__proto__: Map
  ▶__proto__: Object
  path: "QmWpyxSRV9H56qBMYuMSzt6JZZxsLmoM6ouigPsLbfa6Uf"
  size: 52
 ▶__proto__: Object
```

Figure 4. File Stored on IPFS

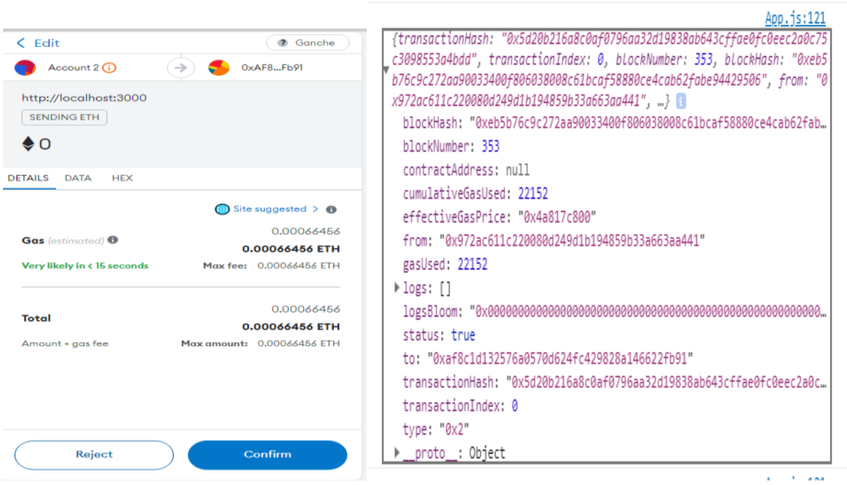
Algorithm 3 Storing the file in IPFS

```
Input : Image file or document by manufacturer
Output : CID of the uploaded file
 begin
   is file uploaded:
       apply AES encryption to file
       store the file in IPFS
   return the CID of the file
```

3. 8. Blockchain Transaction

A blockchain transaction refers to the process of transferring value or data from one party to another on a blockchain network. The CID of the file is stored in block as a transaction. While Storing the data in the blockchain it will provide a transaction hash and block number. The stored data can be identified using Block Number and Transaction hash. A transaction hash is a unique identifier that is assigned to each transaction that occurs on the blockchain. It is a string of alphanumeric characters that is generated using a cryptographic hash function, which takes the transaction data and produces a fixed-length output.

1. Adding an IPFS hash as a transaction in a blockchain requires the following steps:
2. Convert the IPFS hash into a format that can be stored in the blockchain. One common method is to use the hexadecimal representation of the hash.
3. Create a transaction with the IPFS hash as the data payload.
4. Submit the transaction to the blockchain network. This typically involves broadcasting the transaction to a network of nodes, which will validate the transaction and add it to the blockchain ledger.
5. Wait for the transaction to be confirmed.
6. Once the transaction is confirmed, the IPFS hash will be stored in the blockchain ledger, along with the other transaction details.



http://localhost:3000
SENDING ETH
0

Gas (estimated) 0.00066456
0.00066456 ETH
Very likely in < 15 seconds Max fee: 0.00066456 ETH

Total 0.00066456
0.00066456 ETH
Amount + gas fee Max amount: 0.00066456 ETH

```
{transactionHash: "0x5d20b216a8c0af0796aa32d19838ab643cffae0fc0eec2a0c75
c3098553a4bdd", transactionIndex: 0, blockNumber: 353, blockHash: "0xeb5
b76c9c272aa90033400f806038008c61bcaf58880ce4cab62fabe94429506", from: "0
x972ac611c220080d249d1b194859b33a663aa441", …}
  blockHash: "0xeb5b76c9c272aa90033400f806038008c61bcaf58880ce4cab62fab…
  blockNumber: 353
  contractAddress: null
  cumulativeGasUsed: 22152
  effectiveGasPrice: "0x4a817c800"
  from: "0x972ac611c220080d249d1b194859b33a663aa441"
  gasUsed: 22152
 ▸logs: []
  logsBloom: "0x000000000000000000000000000000000000000000000000000000…
  status: true
  to: "0xaf8c1d132576a0570d624fc429828a146622fb91"
  transactionHash: "0x5d20b216a8c0af0796aa32d19838ab643cffae0fc0eec2a0c…
  transactionIndex: 0
  type: "0x2"
 ▸__proto__: Object
```

Figure 5. (a) and (b) Transaction validation and Confirmation of Product

Algorithm 4 Storing the CID in Blockchain

```
Input : CID of the uploaded file
Output : Transaction hash
 begin
     login with metamask wallet
     Request to store the CID in block
     Send the notification for confirmation
     is accepted
        store the CID in block
     else
        send a notification stating request failed
   return the transaction hash
```

3. 9. Proof of Fair Chance

Proof of Fair Chance (PoFC) is a distributed consensus algorithm designed to ensure that all participants in a blockchain network have a fair chance to participate in the consensus process. Designing a consensus algorithm with multiple constraints requires a careful balance between security, decentralization, and practical considerations. The following are constraints used in the Proof of Fair Chance algorithm.

- Minimum computing power: To participate in the consensus process, nodes must have computing power that meets a minimum threshold. This helps to ensure that nodes are contributing enough computational resources to the network to help secure the blockchain.
- Minimum balance: To participate in the consensus process, nodes must also have a minimum balance of cryptocurrency. This helps to ensure that nodes have a stake in the network and are incentivized to act honestly.
- Maximum number of previously mined blocks: To prevent any one node from dominating the consensus process, nodes are limited in the number of blocks they can mine consecutively. This helps to ensure that the blockchain remains decentralized and resistant to manipulation.
- Maximum network bandwidth: Nodes are limited in the amount of data they can send and receive over the network. This helps to ensure that the blockchain remains accessible to all nodes, even those with slower network connections.
- Maximum storage capacity: Nodes are limited in the amount of storage they can allocate to the blockchain. This helps to ensure that the blockchain remains lightweight and accessible to all nodes.

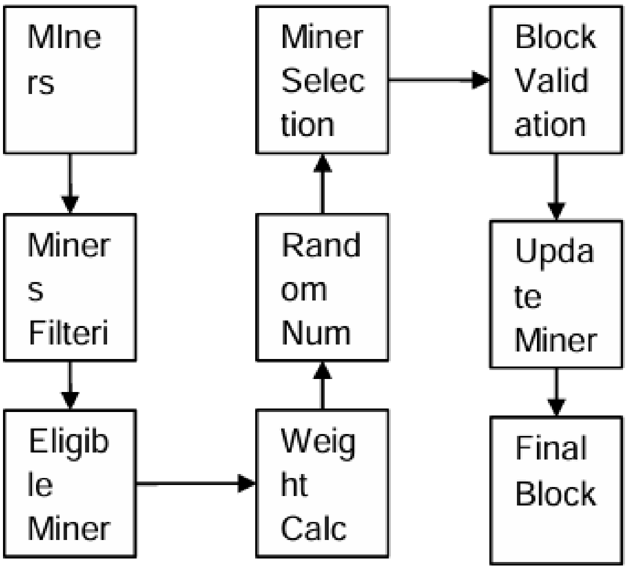


Figure 6. Block diagram for Proof of Fair Chance (PoFC) consensus algorithm

Figure 6 show the block diagram for Proof of Fair Chance consensus algorithm with the following steps:

- Miners: Represented as the initial component, minors participate in the consensus process by contributing computing power and having a stake in the blockchain network.
- Minor Filtering: Minors go through a filtering process to determine eligibility for participating in the consensus based on constraints such as minimum computing power, minimum balance, maximum consecutive blocks, maximum network bandwidth, and maximum storage capacity.
- Eligible Minors: The filtered Minors proceed to the eligible Minors component, representing the Minors who meet the minimum requirements and constraints and are eligible to participate further.
- Weight Calculation: The weights of the eligible minors are calculated based on factors such as stake, reputation, or other criteria. These weights determine the chance of being selected as the block minor.
- Random Number Generation: A random number is generated within the range of 0 to the total weight of the eligible minors.
- Minor Selection: Using the generated random number and the accumulated weights of the eligible minors, a minor is selected for block validation. minors with higher weights have a higher probability of being selected.
- Block Validation: The selected minor is responsible for validating the block, which includes verifying transactions and ensuring consensus rules are followed.

- Update Minor Stats: After successful block validation, the selected minor's statistics are updated. This may include incrementing the numConsecutiveBlocks counter to keep track of consecutive blocks mined by the minor.
- Final Block: The validated block, along with the selected minor, becomes part of the blockchain.

Algorithm 5 Proof of Fair Chance (PoFC) consensus

```
function ProofOfFairChance(block, Minors):
    eligibleMinors = []
    for minor in Minors:
        if minor.computingPower >= minimumComputingPower
           and minor.balance >= minimumBalance
           and minor.numConsecutiveBlocks < maxConsecutiveBlocks
           and minor.networkBandwidth >= maximumNetworkBandwidth
           and minor.storageCapacity >= maximumStorageCapacity:
            eligibleMinors.append(minor)
    totalWeight = 0
    for minor in eligibleMinors:
        totalWeight += minor.weight
    randomNumber = generateRandomNumber()  // Random number between 0 and totalWeight
    accumulatedWeight = 0
    selectedminor = None
    for minor in eligibleMinors:
        accumulatedWeight += minor.weight
        if accumulatedWeight >= randomNumber:
            selectedminor = minor
            break
    block.minor = selectedminor
    selectedminor.numConsecutiveBlocks += 1
    return block
```

In this pseudocode, we assume that the block object contains the necessary information for the block, such as the transactions and other relevant data. The Minors parameter represents the list of potential Minors for the block. The algorithm first filters the Minors list based on the specified constraints, such as minimum computing power, minimum balance, maximum consecutive blocks, maximum network bandwidth, and maximum storage capacity. The eligible Minors are stored in the eligibleMinors list. Next, the algorithm calculates the total weight of all eligible Minors. The weight represents the chance of being selected as the block minor and can be based on factors such as stake, reputation, or other criteria. A random number between 0 and the total weight is generated. The algorithm then iterates through the eligible Minors and accumulates their weights until the accumulated weight surpasses or equals the generated random number. The minor whose accumulated weight crosses the random number threshold is selected as the minor for the block. Finally, the selected minor is assigned to the block.minor attribute, and their numConsecutiveBlocks counter is incremented to keep track of consecutive blocks mined by the minor. The modified block is returned as the result.

3.10 Hashmap creation

Hashmap is a data structure that allows for efficient lookup, insertion, and deletion of key-value pairs.Hashmap is also used to efficiently store and retrieve data. Each transaction is identified by a unique transaction hash , and its sender, receiver, address of the user who has access ,name for the transaction hash and the file name are stored as a value in the hashmap. The hashmap is updated every time a transaction is executed, and the new details are then stored on the blockchain.

To create a hashmap for storing transaction address and IPFS hash using a smart contract, you can follow these steps:

- Define a structure that will represent the transaction details such as transaction hash, sender address, file name, IPFS hash, AES key.
- Declare a mapping that will store the transaction address and its associated data as follows:
- Create a function that will add a new transaction to the hashmap. This function should take in the necessary parameters such as sender, receiver, amount, timestamp, IPFS hash, create a new transaction object, and add it to the hashmap.

Algorithm 6 HashMap Creation

```
struct Transaction {
   address sender;
   string fileName;
   string ipfsHash;
   string aesKey;
}
mapping(address => Transaction) public transactionMap;
function addTransaction(address _sender, string memory _fileName, string memory _ipfsHash,
string memory _aesKey)
public {
   Transaction memory newTransaction = Transaction(_sender, _fileName, _ipfsHash,
_aesKey);
   transactionMap[_sender] = newTransaction;
}
```

Table 1. Hash Map for transaction details

Transaction hash	Sender address	File Name	IPFS hash	AES Key
0xb1e7a6fad751828468d152435990e97f26e6bdaa3a3d2f52219976ca54b25	0×972AC611c2200800249d1B194859833A663Aa441	file1.doc	QmT5NvUtoM5nWFfrQdVrFtvGfKFmG7AHE8P34isapyhCxX	QeThWmYq3t6w9z$C
0xf74bfec61f12d61a4ac7bbe0f4d58b10c318720d34a81b8aadf06f849ad9c74d	0xf4b176b174204564D2edD1dA272e15C47FBD3CFd	Image1.png	QmRBkKi1PnthqaBaiZnXML6fH6PNqCFdpcBxGYXoUQfp6z	PdSgVkYp3s5v8y/B

3.11 Name server creation

To create a name server smart contract for transaction hash and name, sender/owner address and name, you can follow these steps:

- Define a struct that represents a name record, which includes the transaction hash, owner/sender address, and name.
- Declare a mapping that will store the names and their associated records.
- Create a function that allows users to register a name by providing the transaction hash, their address, and the name they want to register.
- Create a function that allows the owner of a name to update the transaction hash associated with it.
- Create a function that allows anyone to lookup a name and retrieve its associated record

Algorithm 7 NameServer Creation

```
struct NameRecord {
    bytes32 transactionHash;
    address owner;
    string name;
}
mapping(string => NameRecord) public nameRegistry;
function registerName(bytes32 _transactionHash, string memory _name) public {
    require(nameRegistry[_name].owner == address(0), "Name already registered");
    NameRecord memory newRecord = NameRecord(_transactionHash, msg.sender, _name);
    nameRegistry[_name] = newRecord;
}
function updateTransactionHash(string memory _name, bytes32 _newTransactionHash) public {
    require(nameRegistry[_name].owner == msg.sender, "Only the owner can update the
transaction hash");
   nameRegistry[_name].transactionHash = _newTransactionHash;
}
function lookupName(string memory _name) public view returns (bytes32, address, string
memory) {
    NameRecord memory record = nameRegistry[_name];
    return (record.transactionHash, record.owner, record.name);
}
```

Table 2. Name Server for transaction hash, sender and receiver address

Transaction hash	**Product Identifier Code**
0×972AC611c2200800249d1B194859833A663Aa441	PID001
0xf74bfec61f12d61a4ac7bbe0f4d58b10c318720d34a81b8aadf06f849ad9c74d	PID010

3.12 Product Verification

Verification of a product is the process of checking the authenticity of the product using the unique identifier. This module can be done by scanning the QR code to view the product details. First the invisible QR code on the product to be scanned and the product details will be displayed.The product id will be used to retrieve the transaction hash from the name server. which will be used to retrieve the CID of the image file. The CID obtained from the blockchain , used as an identifier to obtain the stored product details along the generated QR code from the IPFS.The retrieved product information is compared to the physical product to ensure that they match.Next, the file is retrieved from the IPFS system and encrypted using AES encryption to ensure confidentiality. The user is then able to decrypt the file using a key provided to them by the smart contract.This includes checking the product name, manufacturer details, product serial number, and product price.Once the product information has been verified, the system will confirm the authenticity of the product. If the product information matches the physical product, the product is considered genuine.

3.13 Access control verification

The steps involved in access control verification are as follows:

- A user requests access to a file stored on a blockchain network.
- The smart contract responsible for access control verifies the user's credentials to confirm that they have the required permissions to access the file.
- If the user has the necessary permissions, the smart contract allows access to the file and retrieves the CID from blockchain transactions.
- If the user does not have the required permissions, the smart contract denies access to the file and may inform relevant parties about the unauthorized attempt to access the file.

3.14 Blockchain Retrieval

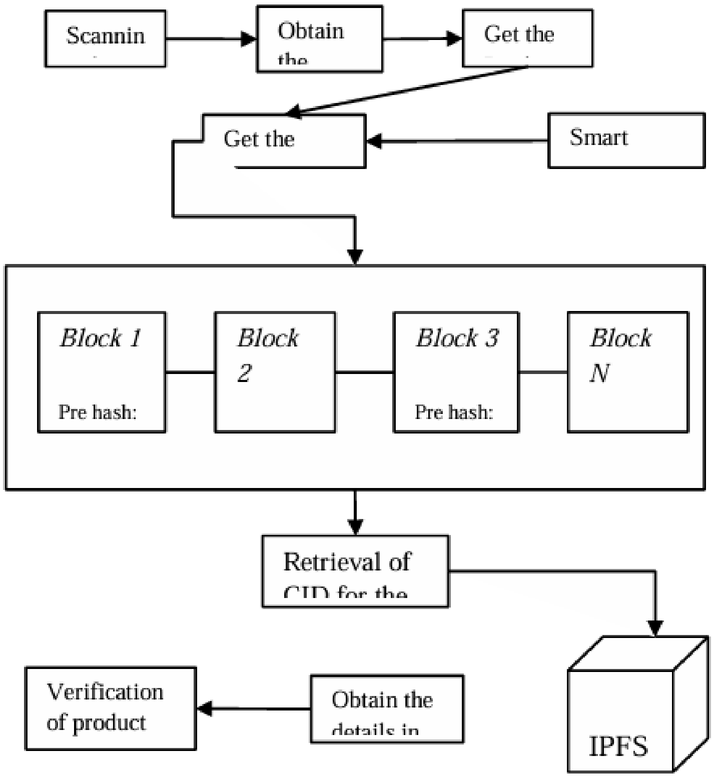


Figure 7. Retrieval Architecture of the proposed system

The file can be retrieved from the blockchain by using Transaction hash. When the user obtains the transaction hash, then it is used to retrieve content id from hashmap.

3.15 IPFS retrieval

Retrieving data from IPFS involves locating and downloading the files from the IPFS network using the unique identifier known as the Content Identifier (CID). CID can be retrieved from the blockchain, it will be used to obtain the encrypted file from the IPFS .

- Retrieve the encrypted file from IPFS: Use the IPFS command IPFS get to retrieve the encrypted file from IPFS using the hash that was generated when you added the file to IPFS.
- Decrypt the file: Use the same AES encryption tool that you used to encrypt the file to decrypt the file and access its contents.

3.16 AES Decryption

Now the file obtained from the IPFS is an encrypted file that should be decrypted in order to get a readable file. The process in AES decryption is as follows:

- To decrypt AES encrypted data stored in IPFS, the encrypted data must first be obtained from the IPFS network using the Content Identifier (CID).
- Next, the decryption key used to encrypt the data must be obtained from the hash map.
- The data can then be decrypted using an AES decryption algorithm.
- After decryption, it is important to verify that the decrypted data matches the expected data to ensure that it has not been tampered with.
- As a result, even if someone attempts to access the encrypted file by obtaining the file hash from the public blockchain, she will not be able to decrypt the file without the key hence protecting the privacy in the process.
- After decryption, the original file will be displayed

4. DISCUSSION

The performance metrics for the IPFS network are discussed below:

Table 3. Performance metrics for the IPFS network

Performance Metric	Description	Target	Value
Storage capacity	The maximum amount of data that can be stored in the IPFS-based blockchain storage network	Scalable and adjustable based on demand	1 PB
Latency	The time taken for a request to reach the IPFS network and for the data to be retrieved	Low latency, preferably less than 1 second	750 ms
Bandwidth	The amount of data that can be transmitted over the network in a given time	High bandwidth, preferably greater than 1 Gbps	2 Gbps
Security	The level of security provided by the IPFS-based blockchain storage network	High security with strong encryption and secure nodes	AES-256 encryption, secure node architecture, regular security audits
Cost	The cost of using the IPFS-based blockchain storage network	Low cost with minimal transaction fees and affordable storage prices	$0.10 per GB per month

Table 4. Performance metrics:IPFS network Vs. Blockchain

Performance Metric	IPFS	Blockchain
Retrieval Speed	Faster due to distributed hash table (DHT) and parallel downloads from multiple nodes	Slower due to verification and consensus, and limited by block size
Availability	Files can be cached locally, increasing availability	Files may not be available on all nodes, reducing availability
Reliability	Data integrity can be verified using content-addressing and cryptographic hashes	Data integrity is ensured through consensus mechanisms and blockchain structure
Cost	Retrieval is generally less expensive than storage	Retrieval can be expensive due to transaction fees and computing power required for verification
Ease of Use	Easier to use, can be accessed through web browser or command-line interface	Requires specialized knowledge of blockchain technology and a dedicated wallet
Scalability	More scalable due to distributed architecture	Can be limited by block size and number of nodes
Security	Relies on encryption and distributed storage for security	Relies on consensus mechanisms for security

Table 5. Performance metrics:Invisible QR code Vs. Normal QR Code

Visibility	Nearly invisible, blends with surroundings	Visible, distinct black squares
Aesthetics	Does not disrupt object/surface design	Can be visually obtrusive

Scanning Compatibility	May require specialized scanners or specific lighting condition	Compatible with standard QR code scanners
Readability	May be challenging under poor lighting or with low-quality scanners	Readable under various conditions
Security	Can embed secret information for added security	Only contains public information
Durability	Resistant to wear, tear, and environmental factors	Susceptible to damage or degradation
User Awareness	Requires user education on the existence and functionality	Familiar to most users
Implementation Cost	May involve additional costs for specialized inks and equipment	Standard QR codes are cost-effective
Integration	Requires adjustments to existing processes and workflows	Easily integrated into existing systems
Application	Useful for discreet labeling and maintaining aesthetics	Commonly used for various purposes

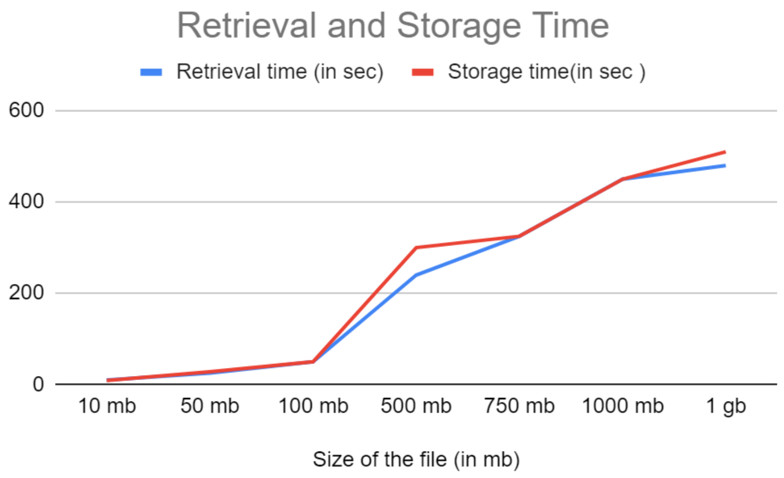


Figure 8. Retrieval and Storage Time based on size of the file.

Figure 8 shows the relationship between size of the file , storage and retrieval time.The storage and retrieval time for the file will be based on the size of the file. As the file size increases , the time taken for storing and retrieving the file will also increase.For large file , the storage time will be quite large.

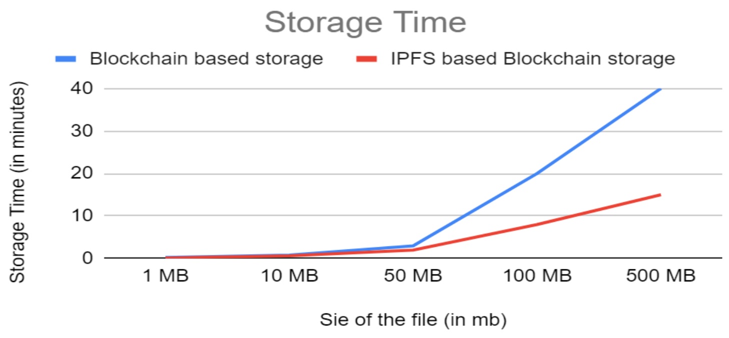


Figure 9. Storage time for both Blockchain and IPFS based Blockchain based on file size

Figure 9 shows the comparison of storage time needed for Blockchain Storage and IPFS Based Blockchain storage based on the size of the file.The storage time needed for IPFS based blockchain is less than the Blockchain based storage . Therefore IPFS based blockchain storage takes minimum time to store a file.

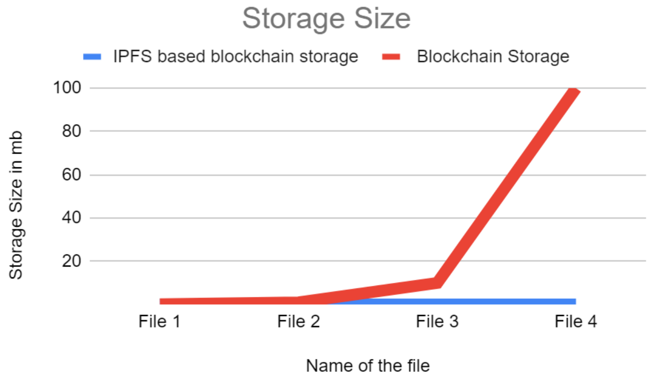


Figure 10. Storage size for both Blockchain and IPFS based Blockchain

Figure 10 shows the Storage size needed for Blockchain storage and IPFS based Blockchain was compared. The storage size of IPFS based will be very small when compared to the Blockchain based storage.

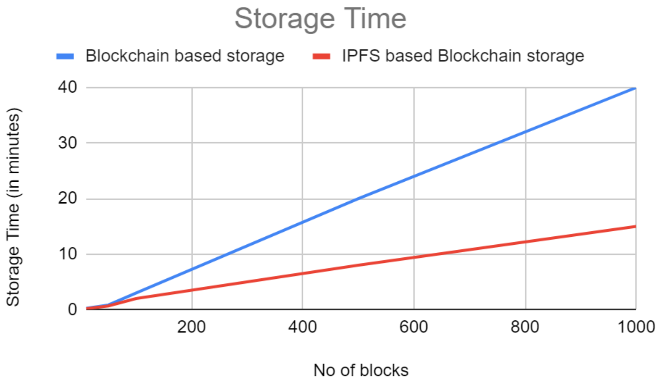


Figure 11. Storage time for Blockchain and IPFS based Blockchain based on no of blocks

Figure 11 shows the comparison of storage time needed for Blockchain Storage and IPFS Based Blockchain storage based on the number of blocks .The storage time needed for IPFS based blockchain is less than the Blockchain based storage . Therefore IPFS based blockchain storage takes minimum time to store a file even if there are more blocks.

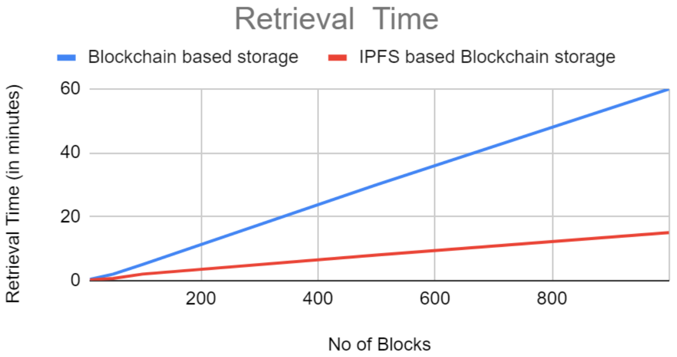


Figure 12. Retrieval time for Blockchain and IPFS based Blockchain based on no of blocks

Figure 12 shows the comparison of retrieval time needed for Blockchain Storage and IPFS Based Blockchain storage based on the number of blocks .The time taken to retrieve a file from IPFS based blockchain is less than the Blockchain based storage . Therefore IPFS based blockchain storage takes minimum time to retrieve a file even if the file size is large.

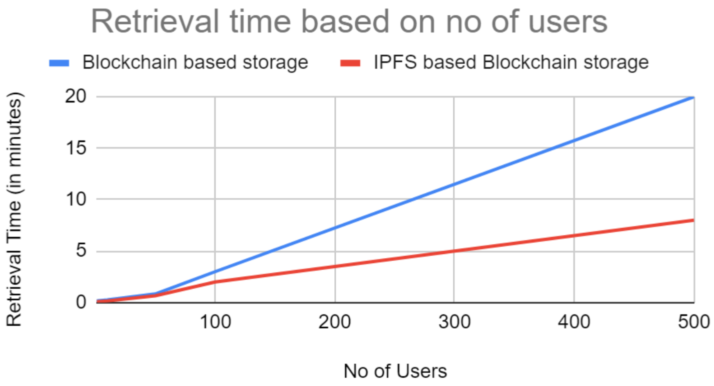


Figure 13. Retrieval time for Blockchain and IPFS based Blockchain based on no of users

Figure 13 represents the relationship between retrieval time based on the number of users for IPFS based Blockchain storage and Blockchain Storage . The retrieval time for the file will be based on the current users of the system . As the number of users increases , the time taken for storing the file will also increase.For IPFS based Blockchain storage , the retrieval time even if there is more number of users , will be less.

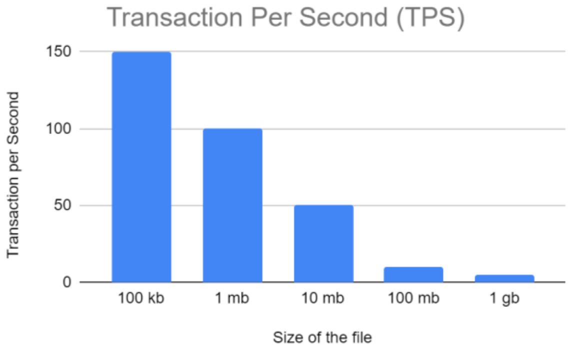


Figure 14. Transaction per second vs File Size

Figure 14 shows transaction per second vs File size. Transaction per second (TPS) is a metric used to measure the number of transactions processed by a system per second. The file size has an impact on the number of transactions per second. The number of transactions processed in a second is proportional to the size of the file.

5. CONCLUSIONS

In conclusion, the utilization of an invisible code stored in IPFS, with its corresponding Content Identifier (CID) stored in a blockchain, along with the use of a hashmap and name server, provides an effective means of verifying the authenticity of a product. This approach ensures that critical product information is securely stored and tamper-proof, while enabling easy access and retrieval when needed.By incorporating an invisible code, the system allows for discreet labeling and seamless integration with the product's design. This enhances the overall aesthetics and user experience without compromising the verification process.Storing the invisible code in IPFS provides a decentralized and distributed storage solution, ensuring the availability and resilience of the product information. The corresponding CID stored in the blockchain acts as a reference, linking the product's unique identifier to its verifiable information.The utilization of a hashmap and name server further enhances the efficiency of the system. The hashmap allows for quick and reliable retrieval of the product information based on its identifier, while the name server ensures a consistent and standardized naming convention for easy identification and management.Overall, this integrated system empowers consumers and stakeholders to easily and reliably verify the authenticity of a product, instilling trust and combating counterfeit practices. It offers a robust and transparent solution that aligns with the demands of modern supply chains and provides a higher level of assurance and confidence for consumers.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., and. formal analysis, X.X.; investigation, X.X.; resources, X.X.; writing—original draft preparation, X.X.; writing—review and editing, Y.Y.; supervision, Y.Y.;. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding

Data Availability Statement

We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical restrictions, a statement is still required. Suggested Data Availability Statements are available in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics.

Conflicts of Interest

The authors declare no conflict of interest

REFERENCES

1. Sun, J., Yao, L., Li, X., & Gao, F. (2018). Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records. Journal of medical systems, 42(8), 136.https://doi.org/10.1109/ACCESS.2020.2982964
2. Ali, M., Shea, R., & Nelson, J. (2016). Blockstack: A Global Naming and Storage System Secured by Blockchain. Proceedings of the 2016 USENIX Annual Technical Conference (USENIX ATC '16), 181-194.
3. Rathore, V., Arora, S., & Jain, S. (2020). A Secure Storage and Retrieval System for Encrypted Data in IPFS using Shamir's Secret Sharing. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-7.
4. Afzal, M., Hussain, W., Farooq, M., & Imran, M. (2021). Secure Data Sharing in IPFS using Blockchain-based Identity Management. 2021 12th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-7.
5. I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, "Blockchain for healthcare data management: opportunities, challenges, and future recommendations," Neural Computing and Applications, vol. 34, no. 14, pp. 11475–11490, Jan. 2021, doi: 10.1007/s00521-020-05519-w.
6. Abdurrashid Ibrahim Sanka, Ray C.C. Cheung(2021) "A systematic review of blockchain scalability: Issues, solutions, analysis and future research" , Journal of Network and Computer Applications Volume 195, 1 December 2021, 103232 https://doi.org/10.1016/j.jnca.2021.103232
7. Masaki Fujikawaa, Yui Uenoa, Daiji Osawab, Eigo Nishimurac, Koutaro Gomid,Hideki Iwasakid, Toshiaki Haradae, and Naoki Adachie "A Method of Firing Nearly Invisible 2D Code with Public and Secret Information on Ceramic Products" , 2019 7th IIAE International Conference on Intelligent Systems and Image Processing , 116-121
8. Zhongpai Gao, Guangtao Zhai and Chunjia Hu (2015) "The Invisible QR Code',23rd ACM international conference on Multimedia , October 2015 , 1047–1050 https://doi.org/10.1145/2733373.2806398
9. Rongjun Chen ,Yongxing Yu ,Jiangtao Chen , Yongbin Zhong , Huimin Zhao ,Amir Hussain and Hong-Zhou Tan (2020) "Customized 2D Barcode Sensing for Anti-Counterfeiting Application in Smart IoT with Fast Encoding and Information Hiding" , Sensors 2020, 20, 4926; https://doi.org/10.3390/s20174926
10. Ren, Y.; Zhu, F.; Sharma, P.K.; Wang, T.; Wang, J.; Alfarraj, O.; Tolba, A. Data Query Mechanism Based on Hash Computing Power of Blockchain in Internet of Things. Sensors 2020, 20, 207. https://doi.org/10.3390/s20010207
11. Pradhan, Singh, Verma, Kavita, Kaur, Roy, Shafi, Wozniak and Ijaz, "A Novel Blockchain-Based Healthcare System Design and Performance Benchmarking on a Multi-Hosted Testbed," Sensors, 2022, 22(9), 3449, https://doi.org/10.3390/s22093449
12. Tijan, E.; Aksentijevi´c, S.; Ivani´c, K.; Jardas, M. Blockchain technology implementation in logistics. Sustainability 2019, 11(4), 1185; https://doi.org/10.3390/su11041185