Computer Science and Information Systems 20(4):1687–1705 https://doi.org/10.2298/CSIS230320061F

Ownership Protection System for Partial Areas on Image Data using Ethereum Blockchain*

Natsuki Fujiwara and Shohei Yokoyama

Department of Computer Science, Faculty of Systems Design,
Tokyo Metropolitan University, Tokyo, Japan
fujiwara-natsuki@tmu.ac.jp
shohei@tmu.ac.jp

Abstract. Our proposed method utilizes blockchain technology to safeguard the ownership of specific regions within image data. In our approach, diverse values could be assigned to each region based on its importance, and only users with ownership rights can access these designated regions. This ensures the protection of ownership rights for individuals in any given region of an image. Identified regions are individually encrypted using an XOR cipher, and a corresponding key image is generated for decryption, thereby preserving the privacy of the encrypted region. Non-fungible tokens (NFTs) are employed to protect the key image and manage the ownership of each object in the image data. The NFT for the key image is generated by the key holder (who possesses the entire image), and the ownership NFT is acquired by the user who needs access to the key NFT. Furthermore, the ownership NFT and the key NFT are verified for a match by the judgment function, and only upon successful validation, the NFT is displayed on the screen. This method enables different values to be assigned to various parts of an image, facilitating the transfer and sharing of ownership. Additionally, the original image's owner can benefit financially based on the value of the image, thus enhancing the overall security of image data.

Keywords: Blockchain, Ownership, NFT.

1. Introduction

Image theft and ownership have skyrocketed with the increasing adaptation of the internet. Thus, ownership protection of image data has become critical. Blockchain is a decentralized database providing secure and transparent protection to records of all kinds. All transactions are recorded and stored in chronological order[1]. Also, blockchain can track and trace all recorded transactions. It is a useful and futuristic technology to improve data management and prove ownership of collected data such as digital art, music, and land. The main features of blockchain are as follows [2] [3]:

- **Immutability**: The ability of a blockchain ledger to remain unchanged, unaltered and indelible by cryptography and distributed consensus mechanisms. Centralized databases can be corrupted and rely on third parties to retain information.

* This is an extended version of the 14th International Conference on Management of Digital EcoSystems.

- **Decentralization**: The shift of control and decision-making from a centralized entity (an individual, organization, or group) to a decentralized network. Decentralization provides several benefits to blockchain network. One of example benefits is security since there is no central point of failure or attack that can be exploited by malicious actors. Also, it ensures scalability and robustness using the resources of participating nodes.
- **Integrity**:Integrity is important to ensure that all data stored on the blockchain is accurate and reliable, with no possibility of manipulation or corruption. Integrity is maintained through a combination of cryptography and decentralized consensus mechanisms. Each block in the chain contains a unique cryptographic hash created using a complex mathematical algorithm. This hash serves as a unique identifier for the block and is generated based on the data stored in the block, making it virtually impossible to alter the data in the block without being discovered by anyone.
- **Anonymity**: Allows anonymous transactions without registering user bank accounts or interacting with traditional financial gatekeepers. To prevent the true identity of the blockchain participants from being known, cryptographic functions are used to disguise or anonymize them, as appropriate. Public-private key cryptography is used for making it possible.

Recently many projects are creating a data management/tracking system, particularly in logistics and supply chain area, with blockchain [4] [5]. For instance, IBM's Food Trust [6] has been developed to enhance visibility and accountability across the food supply chain. Blockchain prevents tampering by allowing unalterable shared records of food origin, transaction details, and processing information. This study focuses on image management rather than developing those types of systems.

Non-fungible tokens (NFT) are commonly used to protect ownership of image data in blockchains. Opensea[1] is the world's first and largest NFT marketplace. It is similar to Amazon, but the listed items are digital collectibles in the form of NFTs. It is common for NFT marketplaces to see the whole image data without holding ownership. In addition, one image equals one ownership, and ownership cannot be split depending on the regions of an image.

NFTs is one of the contributions by smart contracts in Ethereum. NFTs are defined by smart contracts. Some blockchains have functions to execute and verify secure application code called a smart contract. The term smart contract was advocated by Nick Szabo in 1997 [7]. In 2008, bitcoin was proposed by Satoshi Nakamoto [8]. This was the first cryptocurrency based on the revolutionary technology of blockchain which until then had no distributed ledger. Because the blockchain contained only information about the transaction, it is challenging to stipulate the transaction conditions in a new block. Nevertheless, this technology was the catalyst for smart contracts.

Currently there are many different architectures and different user types of blockchain. An example is the Ethereum introduced by Vitalik Buterin [9], making it possible to use smart contracts. Ethereum has expanded upon Bitcoin's basic functions and smart contracts, which help to develop complex applications.

A smart contract is a set of agreements specified in digital form, including a protocol for the parties to execute these commitments. It consists of the value, address, functions,

[1] https://opensea.io/

and state. [10] Smart contracts in Ethereum make it possible to declare digital items as a non-fungible token (NFT) and add data about the creator and owner. Moreover, NFT ownership can be exchanged, transferred, or processed under the rules of smart contracts. A merit of using smart contracts is cost saving. Fees conventionally paid to intermediaries and third parties that guarantee reliability will become redundant. Moreover, the time required for a series of procedures is reduced, and intermediaries will fail to extract information.

We believe our proposed system could be applied to create new services at tourist and photogenic spots. For example, it could be applied to a commemorative photo service at tourist spots. We envision a situation in which a spherical camera is used to take a commemorative photo. The reason why we use the camera is because a 360◦ view of the landscape is shot at one time so that the image provides a sense of realism.

In this situation, the proposed method would be a practical new service that would allow users to purchase and view only specific image areas in which they have ownership rights. The ownership rights held by the system user are only for the user himself, and he does not hold the ownership rights of others. This service protects the privacy of passersby and people who do not want to be seen in the image. If a passerby appears in the photo, the mosaic cannot be removed by anyone who does not have ownership rights. In the future, the system will be integrated with the background scenery in areas where strangers, such as passersby, are present, and the person itself will be removed.

Another applicable case is for an image selling system. When buying a group photo, the current practice is to purchase one group photo. However, with this system, it is possible to give different ownership rights to each person in the image, so if users want images that show everyone in the photo, they must purchase ownership for everyone.

In our research, these specified regions have objects detected by an object detected model. We use You look only once(YOLO) [11] to detect objects. Images captured by spherical cameras are converted to panoramic images (Figure 1). It can be obtained over the entire 360◦ angular range in a plane with one shot. Panoramic images can be used similar to the normal images. The use situation envisioned for this research is a commemorative photo service at theme parks such as tourist attractions and amusement parks. Therefore, taking pictures of the entire scene, rather than a limited background, can create a more realistic feeling of presence, as well as the possibility of augmenting Augmented reality or Virtual reality.

In addition, if using a regular camera such as a smartphone, the scene is fixed, and the camera needs to be turned around to show other scenes. With a 360-degree camera, on the other hand, all scenes can be captured at once. Because of this feature, we thought it would be more efficient to use a 360-degree camera that can capture the entire scene at once, such as at a sightseeing spot, when you want to view the entire scene at once rather than just one spot. In this research, since the purpose of this research is to protect ownership for areas of an image, there is no problem with limited background scenes. In the future, however, we would like to apply this method to protect the ownership of arbitrary areas in video with real time processing. In this case, we believe that it is easier to track people when a larger area is captured efficiently by a single camera. With a camera such as a smartphone, it is necessary to physically orient the camera in relation to the moving person. However, if we used a 360-degree camera, we thought that since the entire background is being captured at once, the camera need only be fixed and tracking of the person can be done easily. This

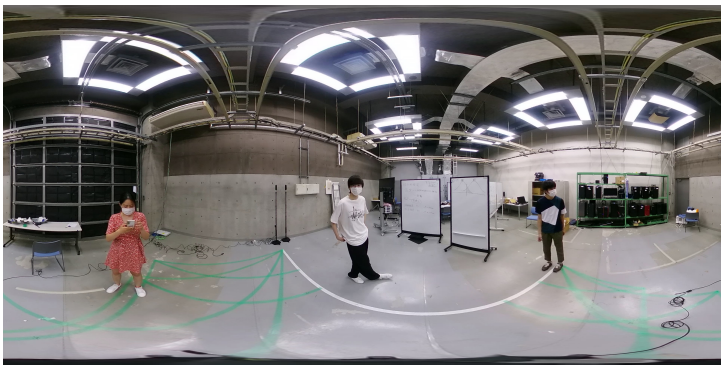


Fig. 1. Panoramic image

is because when tracking a person using a single ordinary camera such as a smartphone with a limited shooting range, if a person enters or exits the frame and then enters the frame again, he or she may be identified as a new person. To prevent this, multiple regular cameras can be installed so that there are no blind spots. However, we believe that using a 360-degree camera allows for efficient tracking, since only one camera captures images in all directions at once and it has less chance to detect the same person as a different person.

We encrypt detected object regions with XOR cipher and create key images corresponding to each area. Encryption converts the original image to a secure unpredictable image. We use the easiest image encryption method since this paper's main purpose is protecting ownership and giving a different value to selected regions of image data. Those key images would be required to decrypt. Crypted areas are shown as mosaic images and objects in this space cannot be identified. Those users having access to the right key images can see the original scenery under encrypted areas that support the key images.

The three main contributions of this study are as follows:

- **Sharability**: Our method makes it possible to give different values to each region and share ownership of a specified area with multiple people on one image data.
- **Affinity** Blockchain is affinity to payment method. Thus, key holders can make financial profit easily. Besides, ownership can be transferred through financial interactions.
- **Security**: Key to decryption is managed safety in the blockchain. They will be unlikely to re-write and leak out.

The remaining paper is organized as follows. The research related to the proposed system is introduced in Section 2. Section 3 describes the proposed method, including image processing and blockchain usage. Section 4 presents the result and outline of our user interface. Finally, Section 5 concludes this paper and reveals the future research directions.

2. Related Work

XOR operations[12], advanced encryption standard (AES) algorithm[13], RSA algorithm [14] and chaos-based algorithm [15] are commonly applied image encryption methods. Irfan et al. [16] suggested using RSA and Chaos-based algorithms. However, securing the image data through encryption may come at the risk of assailant interception in a centralized system. In addition, issues of ownership and copyright of original image data may also occur. To solve these issues, we focus on how and where to store the decryption key after encrypting the image data.

Pink[17] proposes a method to protect the integrity of digital images and verify their ownership. This embeds a digital watermark in the image. This watermark is a unique code that can be used to verify the authenticity and ownership of the image. A digital watermark is generated by combining the content of an image with a secret key to ensure that the image cannot be deleted or tampered without destroying it. To verify the integrity and ownership of an image, the watermark is extracted and compared to the original watermark generated when the image was created. Once the watermarks match, the image is considered authentic and owned by the person who generated the watermark. This method can reveal the owner and confirm whether a circulating image is genuine or not. However, in the proposed method, the encrypted image is leaked but the original image is leaked only to the person who holds the ownership. The emphasis is on protecting the privacy of the people in the image. In addition, our system can guarantee financial benefits to the original image holders.

R.Usha et al. [18] proposes a copyright management system that combines digital watermarking and blockchain technology to protect intellectual property rights. The proposed system uses digital watermarks that embed unique identifiers in digital assets such as images and videos as proof of ownership and copyright. It then leverages blockchain technology to create a distributed ledger of copyright ownership records protected by cryptography and distributed consensus mechanisms. This ledger provides an immutable and transparent record of ownership. It also uses smart contracts to automate copyright registration and licensing. The proposed copyright management system offers several advantages, including improved copyright protection, reduced infringement, and streamlined licensing and royalty management. This approach provides ownership protection for the entire image or video content. However, our research gives ownership respectively only to limited areas that need ownership protection, not to the entire image content. This makes it possible to manage a single image by giving different ownership rights to different areas of the image.

Ali et al.[19] proposes a copyright management system using blockchain and IPFS. The image and the copyright information of the image are registered in IPFS. After registering in IPFS, the hash value is created and registered in the blockchain. After that, when browsing with a web browser using the corresponding IPFS link, if the image is original, the ownership information will be displayed. This method uses IPFS and blockchain to verify whether copyright can be verified in a browser. In our study, after storing image data and other data in IPFS, we made it possible to buy and sell ownership by creating an NFT with IPFS links tied to it.

Waqas et al.[20] proposed securing image encryption based on blockchain. The proposed image encryption method uses the pixel change rate, unified averaged changed intensity, and information entropy analysis. Furthermore, to ensure decryption, each pixel

value of a key image is stored in the blockchain. However, the transaction speed reduces because each pixel data is stored in a blockchain. This research is like our work. However, we created NFT as key image data containing each pixel information to decrypt for improved transaction speed. Khan et al.[21] suggested a safe image-sharing framework using blockchain for medical applications. A list of the key owners with access to important images is stored. Moreover, it is impossible to share ownership depending on areas on the image since the key allows accessing one image or partial areas on the image.

Mohamed[22] suggested an image management system using an Ethereum network. They encrypted whole images, and decrypted keys are stored in the blockchain. Rashid et al. [23] proposed a blockchain-based framework for copyright protection by maintaining the integrity of the original content. Users can download content after verifying the programmed transaction contents. Khan et al.[24] introduced a blockchain system to guarantee the authenticity of stored recordings and allow authorities to verify that footage has not been tampered with. Metadata of images and videos are stored in the blockchain to prevent data forgery.

Wang et al.[25] explain NFTs and how they solve security issues like authenticity, integrity, non-reputability, confidentiality, availability, and authorization. Furthermore, they researched fields that can potentially use NFTs effectively. An example is protecting digital collectibles such as image data. The NFT creators can decide on the transaction contract details, for example, giving ownership and prices, making them an efficient way to manage and protect ownership.

Therefore, we use NFTs for our research because they allow image holders to decide on image value and show digital ownership. Furthermore, NFTs are more secure because of blockchain technology. It is unlikely to tamper with ownership or each image detail because those data are stored in each block and are almost impossible to re-write.

Moreover, most of the above studies focus on ownership protection of full image data. We extended it to facilitate protecting partial areas on images and ownership.

3. Proposed Method

We describe the object detection model and illustrate the image encryption method. In addition, we introduce the blockchain components. This section also explains creating ownership of image data as NFTs and outlines the user interface.

3.1. Spherical cameras

A spherical camera uses multiple lenses to combine all pictures taken by different lenses into a single image. This makes it possible to capture an omnidirectional space with a single camera and a single shoot. The captured images are converted to panoramic images by equirectangular projection before storage. Panoramic images allow dealing with the spherical images as normal images. In the converted image, latitude and longitude lines intersect at right angles and equal intervals. High and low latitude areas cause distortions.

Object Detection by Yolo We adopted YOLO, a state-of-the-art object detection algorithm using neural networks, as the object recognition model. This model outputs the coordinates and type of objects.

Previous traditional object recognition methods have classified each region by identifying multiple candidate object-like regions. However, YOLO does not list candidate object regions but uses a single convolutional neural network for prediction. It is also treated as a regression problem that directly predicts the coordinates and object size by training with a dataset containing the object's location and category. The region of the object is represented by establishing the class it belongs to under the condition that some object is present in a particular region. Moreover, it is faster to process and looks at the entire image, which improves accuracy. We use Yolo verison5 in our study.

In addition, we apply the trained model by Microsoft's Coco dataset. This model can detect 80 classes such as dogs, human beings, and cars. We chose humans because they are often seen around the mid-latitudes of image data, minimizing distortion effects from an omnidirectional camera.

3.2. Image encryption by XOR cipher

This section explains the XOR cipher followed by image encryption using it.

XOR cipher We use detected images and a coordinate list of each detected area to encrypt by XOR cipher. Because image encryption method is outside the scope of this study, we use one of the easiest image encryption methods, the exclusive OR (XOR) cipher, for each pixel value.

XOR is a bitwise operator. The result of xor operation is 1 if the two bits are different. The result is 0 if the two bits are the same. An example of calculating using the XOR operator on numbers 38 and 178 is as follows. The decimal value is first converted to the equivalent binary number. The binary equivalent of 38 and 178 is 10110010 and 00100110, respectively. The XOR operation returns the calculated value as 10010100, which is then converted back to the decimal equivalent 48.

Figure 2 shows an example of the encryption process. In step 1, we calculate the image size. Here, the image size is 2×2 and each RGB value is [55,136,244]. Moreover, we create a key sequence, a random number of the same size as the image to be encrypted. The key sequence is converted to a key image.

Image encryption Step 2 shows the results of the XOR operation on each RGB channel. The RGB values of the step 1 image are extracted. XOR cipher is performed separately for each RGB value and key sequence. For example, when the key sequence is 1, the XOR operation to key sequence and blue channel 244 returns 245. The calculation is repeated for each RGB channel and all pixels.

In step 3, the result of each RGB is merged to create encrypted color images. The XOR operation calculates the key and encrypted image to decrypt each region. The process is repeated for each detected area based on the text file of the object location.

3.3. Blockchain side

This section explains protecting ownership and managing key images using blockchain.

Ethereum We use Ethereum, a decentralized open source blockchain, to create NFTs and events such as transferring ownership under specified rules defined by smart contracts.

Smart contracts are digital self-executing contracts executed based on agreement terms between buyers and sellers. Firstly, we explain calling the deployed smart contract to the front.

Smart contracts are designed using a high-level language such as solidity. Compilation generates a JSON file including dates of both EVM bytecode and an application binary interface (ABI). ABI provides summary information needed to execute smart contracts and retrieve variables to invoke them outside Ethereum or from another contract. Then they are deployed to the blockchain network, and the contract address is published. A smart contract can be called and executed by calling those addresses to the front.

The currency used on Ethereum is called Ether (ETH). A transaction fee called gas fees is required to process and validate transactions, including deploying smart contracts or calling them on the Ethereum blockchain. In our research, we use MetaMask[2] as a wallet for ETH.

We use Hardhat[3] network as the test environment. Hardhat is a development environment for compiling, deploying, and testing smart contracts and debugging Ethereum software. It is possible to develop and connect a local or a specific blockchain network, such as a private blockchain. Moreover, fake ETH can be used to try our application since several accounts have fake 1000 ETH for the test net environment. The Hardhat network is connected to MetaMask in this study.

NFT NFTs serve two purposes. One is for ownership of key images, and another is for NFTs as key images. This method ensures that the key images cannot be accessed without ownership. The key image holder can assign different values to each key.

Our NFTs are designed by Ethereum Request for Comments 721 (ERC-721). This standard represents ownership of NFT. Image data is rarely stored on the blockchain because of the high gas fees. When creating image data as NFT, the data is uploaded to the interplanetary file system (IPFS), a protocol and peer-to-peer network for storing and sharing data in a distributed file system. [26]

[2] https://metamask.io/
[3] https://hardhat.org/

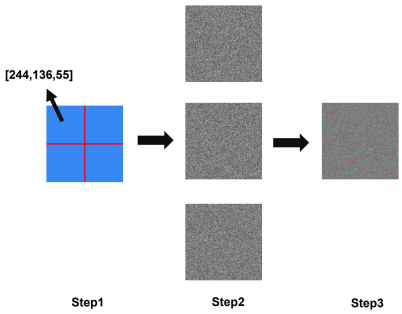


Fig. 2. Encryption method

IPFS relies on a distributed hash table (DHT) to retrieve file locations and node connection information.[27] It is a content address and assigns a unique hash to store files on the network. It also incorporates deduplication technology and is not constrained by a centralized server. In this research, IPFS desktop[4] is used. It assigns a specified ID corresponding to each uploaded image or content. This ID is written in the NFT metadata and stored in the blockchain.

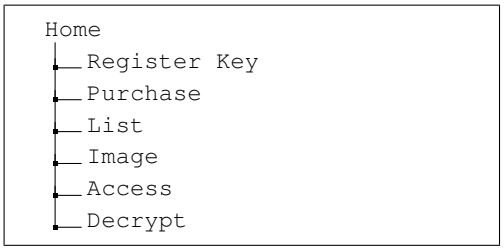
NFTs are defined by smart contracts. We created four smart contracts, two to define the NFTs of ownership rights and key images. The other two specify actions such as obtaining and transferring NFTs.

Details of our smart contracts are as follows: The first two smart contracts have the same content, defining the token type created. This includes a counter function that would tell the number of NFTs created and the mint function. The mint function validates data, creates a new block, and records data into the blockchain. The function checks if the minted token is sent to the right contract to manage NFTs. These two smart contracts can not be converted into one smart contract since Key NFT and ownership NFT tokens are considered as different tokens and need to be defined separately.

The other two contracts define the data type of NFT name, image ID, price, and description as metadata. A smart contract defines two separate events: one for the ownership of NFTs and the other for keyNFTs. Regarding the ownership NFT event, the 'itemcount' value is updated when the ownership NFT is purchased. This value influences the determination and description of whether these contracts also include information about some actions. For example, it is an automatically executed transaction showing from the transfer of NFT ownership. In addition, some regulations to buy are written in the contract; for example, the minimum total amount in the wallet for the buyer is 1 ETH.

User interface Figure 3 shows pages created for different functions using React. We also use the ethers.js library, a complete and compact library for interacting with the Ethereum Blockchain.

Fig. 3. Page content



The home page asks users to connect with MetaMask. Once connected, all transactions are executed by the wallet. On the register key page, the key holder can mint key images and decide the cost of each image. A text box allows entering the name, price, and

[4] https://ipfs.io/

description to be stored in the blockchain. They will be required later to match ownership. On the Purchased page, key images would be added upon its purchase. The list page shows minted key images for key holders. If no key is mint, no image data is found.

The decrypted images are available from the Image page. There is an input box consisting of names and descriptions. This data will be stored in the blockchain and will be used to match ownership with the right key. This page is visible to all users. Ownership can be created here if the user wishes to decrypt each region. On the access page, a list of holding ownership will be shown with the corresponding ownership key images. If there is no ownership holding, they are no image.

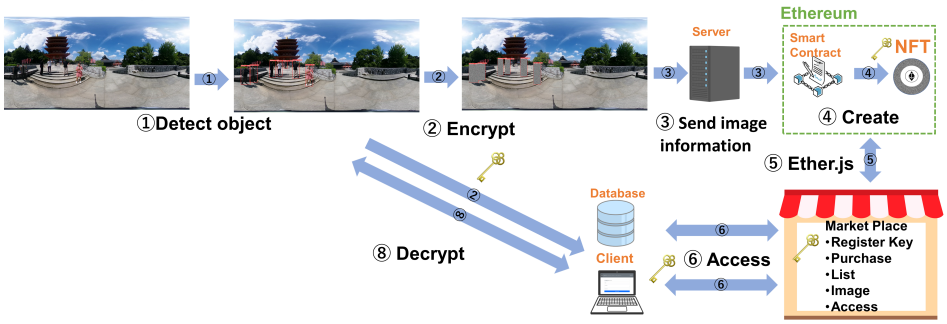


Fig. 4. Outline of our system

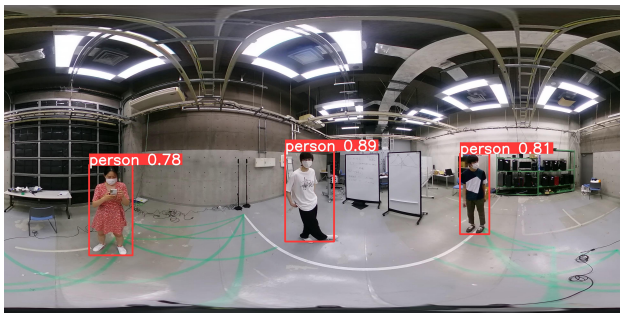


Fig. 5. Detected Object

4. Implementation

Figure 4 shows outline of our system. Server and client are not physically separated at this research. The first half are image processing sections such as detecting objects and

(a) Key **(b)** B **(c)** G **(d)** R **(e)** Merge

Fig. 6. Key and BGR images

encrypting the objects. The latter half shows how to create keys and ownership as NFTs as well as the flow of keys. Keys will move like the figure. Especially in the marketplace, it commands smart contracts as creating ownership of keys and key images. Besides, they ask for transferring ownership. We will describe each detailed process in the following section.

4.1. Object detection

We use a THETA Z1 from Ricoh as input images. The camera lens is focused on the center to avoid the influence of distortion at high and low latitudes when capturing an input image. Figure 5 shows the coordinate file and detected image generated by YOLOv5. Three people were detected accurately. However, objects at high and low latitudes are distorted.

4.2. Encryption

The process images to create an encrypted image for one detected person are depicted in Figure 6. The process is repeated for the other detected two persons. The detected areas are selected based on the coordinate file to create random sequences. The file contains information about the x and y coordinates in the upper left corner of the detected areas and the object's width and height. The size is the same as each detected area. These random sequences are converted into key images with only one channel because the sequences are based on a one-dimensional array. These images will become key NFTs in the next phase.

Besides key image generation, the red, green, and blue channel values are obtained from each object, and the XOR operation is performed for each channel and key image. We merged three encrypted images of each channel into one image and saved them as .png files because they can be compressed without any information loss. Thus, each pixel value can be fully restored when decrypting the images.

The image after successful encryption is shown in Figure 7. We also write ID numbers close to each detected area, such as key0, key1, and key2. These IDs will be used to check whether the held ownership corresponds to the key image.

Figure 8a is a successfully decrypted image with the right key. Figure 8b is decrypted image with the wrong key. When wrong key images are used, the incorrectly decrypted images cannot identify anything. Thus XOR cipher is the easiest way to image encryption and serves at least a minimum function.

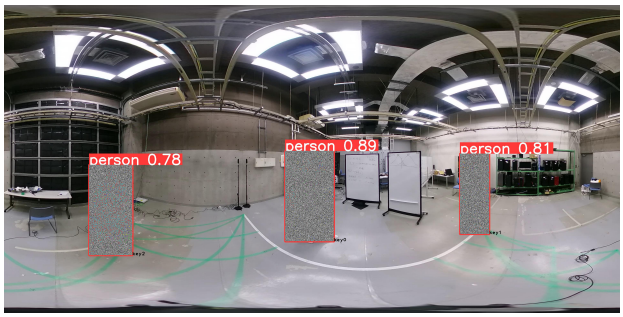
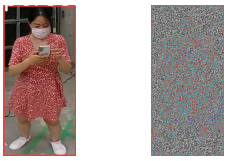


Fig. 7. Encrypted image



(a) Success **(b)** Fail

Fig. 8. Decryption

4.3. Database

During the encryption process, ImgID, KeyID, and the storage path of the encrypted image are registered in the relational database created by MySQL. Figure 9 shows example contents of these tables. ImgID is the image name, and KeyID is the ID assigned in Section4.2. Two tables are created using the ImgID as the primary key. Table accord includes detection coordinates (bbox data), ImgID, and KeyID. The table path is for storing the location of the encrypted image such as Fig.7. Bbox is the x and y coordinates in the upper left corner of the detected areas and the object's width and height. The detection coordinate values, and encrypted image paths stored in the database are utilized for decryption based on the input ImgID and KeyID information.

The first reason for using a database here is that encrypted images and detection coordinates do not require secure management. Secondly, blockchain is not good at inspecting information that has been registered in the past. Lastly, it leads to cost saving since issuing NFT of entire encryption images needs transaction fee. In addition, the reason for using the database and blockchain together this time is to simultaneously prove ownership and search and cite data. The blockchain makes it possible to clarify ownership and makes it difficult to tamper with whether ownership is retained. The database facilitates the search and citation of data registered in the past and enables the encryption and decryption of images.

4.4. WebSocket

In this research, WebSocket, a persistent connection between a client and server, is used for bridging process between Python and JavaScript. Python is used for image processing

Table path

ImgID	Path
0001	moji_0001.png
0002	moji_0002.png

Table accord

ImgID	KeyID	Accord
0001	0	2172 1077 40 91
0001	1	704 895 114 352
0001	2	1026 856 111 300
0001	3	567 931 205 448
0002	0	324 957 160 343
0002	1	145 060 195 380
0002	2	892 868 105 305

Fig. 9. Database table

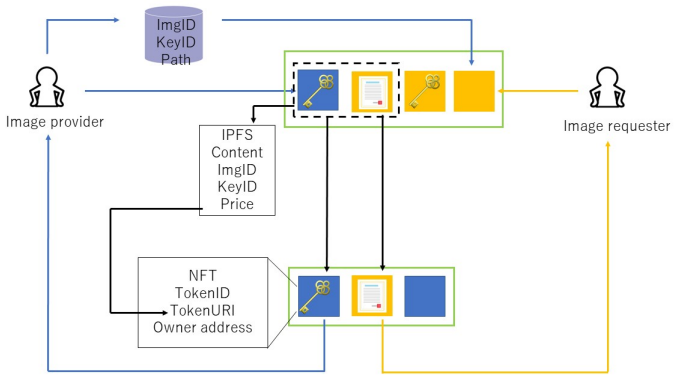


Fig. 10. Register

including object recognition, encryption and decryption. JavaScript is used for building the marketplace and connecting to Ethereum. The Python processing side would obtain related data such as ImgID and KeyID when marketplace asks for image decryption. Then, Python side will send back decryption image to marketplace.

4.5. Data flow

Figure 10 shows the data flow of a key image, key NFT and ownership NFT. This section shows the process by which a key image becomes a key NFT and how the key NFT is disseminated to users.

The image provider is the holder of the original image. The image requester is the user who wishes to obtain the original image. In step 1, the image provider converts the key image to NFT via the marketplace. Step 2 to step 4 is the data flow for the image requester side. The goal during these steps is that encrypted image is decrypted using the key NFT. Step 2 is the ownership creation for acquiring the key NFT. Step 1 and step 2 are the process of creating the NFT. Step 1 and step 2 are storing ImgID, KeyID and image contents which is for step 1 in IPFS.

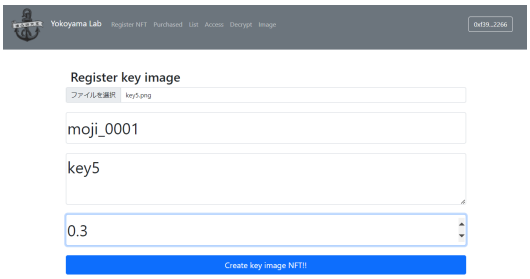


Fig. 11. Register

After the registration to IPFS is successfully completed, a URI is issued. This URI will be included in the metadata of the NFT. The owner address of the person who applied for the issuance of the NFT is also stored. When a key NFT is issued, it is the address of the image provider, and when an ownership NFT is issued, it is the address of the image requester. The address here is the address assigned to the MetaMask in this research.

Step 3 is the purchase of the key NFT. KeyID and ImgID information is obtained from the URI of the ownership NFT and the key NFT. The key NFT can be purchased only when the KeyID and ImgID of the two NFTs match.

In step 4, encryption processing is applied to the target area based on the purchased NFT. The KeyID and ImgID information is obtained from the URI of the purchased key NFT. Once those two IDs are obtained, the IDs are searched in the database and get data of path to an encrypted entire image and bbox. The decryption of the target area is performed based on the entire image and bbox value.

4.6. User interface

We will show the main features of our marketplace. In this research, we made a simple user interface to test the working of the proposed method.

Register Figure 11 is a register page. The function allows key holders to register key images as NFTs and input information to be stored in the blockchain upon completion of the transaction. The input includes name, description, and appropriate price value as information. However, there is no re-write option. Though the price field can vary, the key holders must input the right name (image name) and description (key ID) to access the key image. When successfully registered, the minted images will appear on the List page.

Image The list of partially encrypted images is shown in Figure 12. The locally stored encrypted images are manually added. Moreover, these images do not require storage in a secure environment and are, therefore, not registered as NFTs.

However, this page will create ownership as NFT once the transaction succeeds. A name (image) and description (key ID) are required to get ownership. People requiring

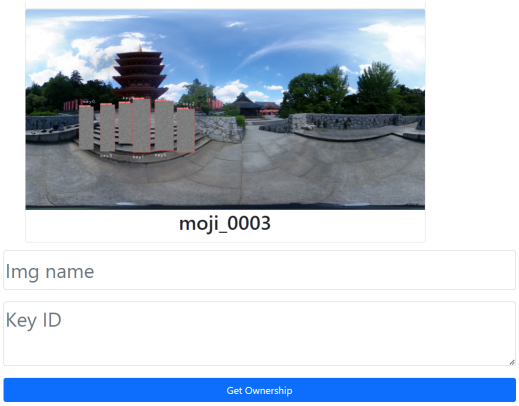
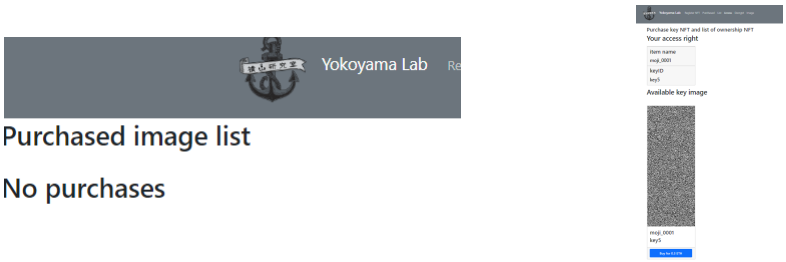


Fig. 12. Image



(a) No Key **(b)** Available key

Fig. 13. Access

ownership must input the name and description following the specific rules. For example, to acquire ownership of key0, the user needs to input img0 as name and key0 as description. The key will not be shown on another page if the wrong value is input.

Access The access page is shown in Figure 13.This page contains the ownership list and key images corresponding to the ownership. No images are screened when no ownership is obtained (Figure 13a). In contrast, key images appear when the name and description in ownership and key NFTs are matched (Figure 13b) using a linear search. The purchased key images are listed on the purchased list page. In this paper, connecting to the decryption page is overlooked. In the future, a new function that sends only purchased images to the decryption function will be designed. When users do not buy the key images, image decryption cannot be achieved.

Decrypt The decryption page is shown in Figure 14. This page shows entire images which obtained ownership areas are decrypted. Currently, the areas that are not owned to access are mosaic-like, but in the future those areas are assimilated with the background so that the photo is natural. For example, for application to video services, assimilation

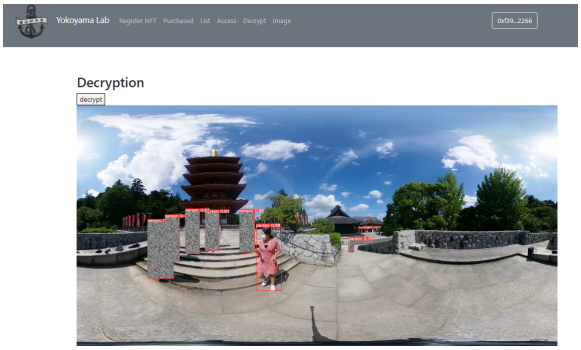


Fig. 14. Decrypt

by difference from the previous frame is being considered, and for difference in application to image services, assimilation methods that tconsider surrounding pixels are being considered.

5. Conclusion and Future Work

We proposed an ownership protection method for limited image regions by assigning different values to each area using blockchain. With this method, we achieve the following conditions.

- **Dividing**: Ownership to one image can be split corresponding to areas and each ownership can be shared with multiple people since users can apply for ownership on the image page. Besides, we can manage which key can be open to users on the register page.
- **Affinizing**: Key image holders can get financial profit by Eth every time a user buys the image. Besides, this earned Eth can be used to buy other keys or use different platforms. Moreover, as a merit for key holders, key holders can decide the value; for example, when the area is more valuable, key holders can set a higher price.
- **Protecting**: Ownership is controlled safety. This method prevents image theft because only limited people having ownership can access the key images to decrypt. The blockchain stores the key images, excluding the third party for saving images. Image ownership is protected because programmed smart contracts transfer ownership.

However, further improvements in usability and security are required. The discussion in this study centered on how to trade ownership. Considering that resale or hacking of keys can occur, we would like to move in the direction of using this system as proof of ownership. When key images are circulated via screenshots, etc., it is possible to determine if the owner is the correct owner by asking whether the ownership NFT is present. Ultimately, the goal is to allow users who wish to use the commemorative images/video

service at tourist spots to send signals to the proposed method's system on their smartphones. And it allows users to purchase ownership rights to their own areas and obtain images/videos. In the future, the following aspects will be considered.

- **Image Encryption methods**: Other image encryption methods might be considered for more secure operation. Though we generated a random sequence by a random function, it might predict rules since the random numbers are pseudorandom. Since the XOR operation used is very simple, we are considering the use of other cryptographic methods such as AES ciphers and the use of JPEG-style layered structures.
- **Matching algorithms**: This study uses a linear search algorithm. Other algorithms, such as binary search algorithm, with improved calculation speed and less memory usage will be used to match ownership and key.
- **key management**: We manually input an partial encrypted image data on image page. We are thinking of using database to save all the images and automatically call it to the front side. The images are not necessary to be stored in blockchain. It dose not need to refer to security aspect since all important regions are concealed. Besides, it leads to cost saving since storing data in blockchain costs Eth.
- **Video application**: We have examined whether the proposed method can provide ownership protection for a portion of an image. In the future, we are considering applying the same process to video type. It is necessary to consider how to manage keys when processing for video. In this research, since it was an image, a new key was issued each time. However, the same key should be used to encrypt the same person on all frames in the case of video since it is better for users to purchase only one key for decryption.
- **New network**: All images need to be reset after testing in the local blockchain. Therefore, we aim to create a private Ethereum blockchain using Geth. Geth is Go Ethereum's[5] standalone CLI client implemented in Go Language. Go Ethereum is one of the three original implementations of the Ethereum protocol.
- **Token standard**: Though ERC721 is common for NFTs, several smart contracts need to be created. However, by combining one smart contract with ERC1155, gas fees can be reduced because the contracts can contain some definition of tokens.
- **Real-time processing**: In the future, we aim to process all steps in real-time. YOLOv5 can immediately detect objects from the input images sent from a camera. However, the delay in sending real-time images to the blockchain and creating each NFT should be investigated.
- **Improving user interface**: With current user interface, a user and key holder need to input image name and key ID by hand. This is not a user-friendly interface. Because they have to input one by one. Besides, there might be a mistake to type wrong text and create incorrect NFTs. To avoid human error and improve user experience, we will change the interface. For example, Name and key ID are automatically input when a user clicks the desire areas.

[5] https://geth.ethereum.org/

– **Mobile access**:In this study, MetaMask login is done using the chrome extension. Currently, it is not possible to log in using the extension from a smartphone browser. Therefore, a login page for smartphone users will be created in the future to allow access from smartphones.

References

1. Wang, H., Zheng, Z., Xie, S., Dai, H.N., Chen, X.: Blockchain challenges and opportunities: a survey. International Journal of Web and Grid Services 14, 352 – 375 (10 2018)
2. de Haro Olmo, F., Varela Vaca, A., Álvarez Bermejo, J.: Blockchain from the perspective of privacy and anonymisation: A systematic literature review. Sensors 20 (12 2020)
3. Kumar, N.M., Mallick, P.K.: Blockchain technology for security issues and challenges in iot. Procedia Computer Science 132, 1815–1823 (2018), https://www.sciencedirect.com/science/article/pii/S187705091830872X, international Conference on Computational Intelligence and Data Science
4. Saberi, S., Kouhizadeh, M., Sarkis, J., Shen, L.: Blockchain technology and its relationships to sustainable supply chain management. International Journal of Production Research 57, 1–19 (10 2018)
5. Pal, K., Yasar, A.U.H.: Internet of things and blockchain technology in apparel manufacturing supply chain data management. Procedia Computer Science 170, 450–457 (01 2020)
6. IBM: About ibm food trust (2019), https://www.ibm.com/downloads/cas/8QABQBDR
7. Szabo, N.: Formalizing and securing relationships on public networks. First Monday 2(9) (Sep 1997)
8. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (Oct 2008)
9. Buterin, V.: Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform (2014)
10. Bahga, A., Madisetti, V.: Blockchain platform for industrial internet of things. Journal of Software Engineering and Applications 09, 533–546 (01 2016)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection (May 2016)
12. Han, J., Park, C.S., Ryu, D.H., Kim, E.S.: Optical image encryption based on XOR operations. Optical Engineering 38(1), 47 – 54 (1999)
13. Zhang, Q., Ding, Q.: Digital image encryption based on advanced encryption standard (aes). In: 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC). pp. 1218–1221 (2015)
14. Sahoo, A., Mohanty, P., Sethi, P.C.: Image encryption using rsa algorithm. In: Udgata, S.K., Sethi, S., Gao, X.Z. (eds.) Intelligent Systems. pp. 641–652. Springer Nature Singapore, Singapore (2022)
15. Guan, Z.H., Huang, F., Guan, W.: Chaos-based image encryption algorithm. Physics Letters A 346(1-3) (Oec 2015)
16. Irfan, P., Prayudi, Y., Riadi, I.: Image encryption using combination of chaotic system and rivers shamir adleman (rsa). Computer Applications 123(6) (Aug 2015)
17. Wong, P.W.: A watermark for image integrity and ownership verification. In: Image Processing, Image Quality, Image Capture Systems Conference (1998)
18. Meng, Z., Morizumi, T., Miyata, S., Kinoshita, H.: Design scheme of copyright management system based on digital watermarking and blockchain. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). vol. 02, pp. 359–364 (2018)
19. Ali Muwafaq, Alchaqmaqchee, a.S.N.A.: Design scheme for copyright management system using blockchain and ipfs. Computer and Degital Systems 10(1) (2021)

20. Khan, P.W., Byun, Y.: A blockchain-based secure image encryption scheme for the industrial internet of things. Entropy 22(2) (2020), https://www.mdpi.com/1099-4300/22/2/175
21. Jabarulla, M.Y., Lee, H.N.: Blockchain-based distributed patient-centric image management system. Applied Sciences 11(1) (2021), https://www.mdpi.com/2076-3417/11/1/196
22. Dedge, O., Shingade, R., Jagtap, A., Yadav, A., Kamble, A.: Image copyright protection system using blockchain. BULLETIN MONUMENTAL 21(2) (2021)
23. Rashid, Mamunur, M., Lee, S.H., Kwon, K.R.: Blockchain technology for combating deepfake and protect video/image integrity. Journal of Korea Multimedia Societ 24(8) (2021)
24. Khan, P.W., Byun, Y.C., Park, N.: A data verification system for cctv surveillance cameras using blockchain technology in smart cities. Electronics 9(3) (2020)
25. Wang, Q., Li, R., Wang, Q., Chen, S.: Non-fungible token (nft): Overview, evaluation, opportunities and challenges 123(6) (Oct 2021)
26. Benet, J.: Ipfs - content addressed, versioned, p2p file system (07 2014)
27. Steichen, M., Fiz Pontiveros, B., Norvill, R., Shbair, W., State, R.: Blockchain-based, decentralized access control for ipfs (07 2018)

Natsuki Fujiwara is a master's student of computer science at Tokyo Metropolitan University. Her work focuses specifically on managing the ownership of image/video data on blockchain.

Shohei Yokoyama is an associate professor at Tokyo Metropolitan University, a visiting associate professor at the National Institute of Informatics, and a research fellow at the University of Tokyo, Japan. He received a Ph.D. degree in engineering from Tokyo Metropolitan University in 2006. As a postdoctoral fellow, he joined the National Institute of Advanced Industrial Science and Technology in 2006. From 2008 to 2018, he was an assistant professor, a lecturer, and an associate professor at Shizuoka University. Since 2018, he has been an associate professor at Tokyo Metropolitan University. His work focuses specifically on data engineering and data science. His research interests include data communication on 5G networks and geo-social big data analysis.

Received: March 20, 2023; Accepted: August 10, 2023.