B-Spot: Blockchain and Steganography based Robust and Secure Photo Transmission Mechanism

Dongchi Li and Pushpendu Kar*

University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo 315100, China
E-mail: scydl1@nottingham.edu.cn; pushpendu.kar@nottingham.edu.cn
**Corresponding Author*

Received 10 November 2021; Accepted 22 March 2022;
Publication 05 July 2022

Abstract

In this paper, a Steganography and Blockchain based robust and secure photo transmission mechanism, named *B-Spot*, is proposed. At the sender side, firstly, a 3-3-2 LSB image steganography algorithm is used to hide a secret photo into a cover photo. Therefore, the existence of the secret photo is concealed. Then the stego-image is broken down into pixels and embedded into blocks. Blocks are connected by the hash values forming a blockchain. Any modifications on the blocks are reflected by the breakage of the chain, which makes the mechanism tamper-evident. Another copy of the blockchain is stored in a hash table for the latter recovery process. The blockchain and the hash table can be transmitted via any network. When the receiver receives the data, it firstly executes the verification process to check integrity of the blockchain. Then the lost and tampered blocks are recovered by referring to the hash table, which makes the mechanism more robust to noises. Finally, the stego-image is reconstructed from the recovered blockchain. Then the receiver can obtain the secret photo following the

Journal of Mobile Multimedia, Vol. 18_6, 1677–1708.
doi: 10.13052/jmm1550-4646.18610

extraction algorithm. The simulation results show that the proposed mechanism has high data capacity, better imperceptibility, reasonable computing time, and strong robustness to noise. This mechanism adds an extra layer of security and robustness to the existing schemes.

Keywords: Blockchain, steganography, robustness, security, photo transmission mechanism, verification, recovery.

1 Introduction

In recent years, network technology has developed rapidly, bringing people many conveniences, such as high-speed file transmission and easier information acquisition. Due to the easy access to the open network and the Internet, the security of digital information, such as digital photos, has increasingly become a significant issue [27]. According to Cisco,s report [3], 46% of workers exchanging records between their work and individual computers. If the transmission network used is insecure, most companies' sensitive data is in danger of being leaked. As reported by statista.com [11], nearly 1500 data breaches happened in 2019 in the United States, with over 160 million secret records exposed. Therefore, a solution to protect us from these network security threats is urgently needed. In this paper, a blockchain and steganography based photo transmission mechanism is proposed, for secure and robust image transmission in the network.

Steganography is the art of hiding secret information by concealing it into other cover files. Sated by Johnson and Jajodia 'Steganography' is derived from Greek, which literally means 'covered writing' [12]. The characteristics of steganography is the existence of secret data becomes hard to identify, which is a very effective way to improve the security of the secret file of the image type. Commonly used image steganography methods can be divided into four categories: spatial domain methods, transform domain technique, distortion techniques, and masking and filtering [9]. In the proposed method, a famous spatial domain method, the least significant bit (LSB), is used to hide the secret image. Using the LSB method, the embedding and extraction process becomes simple, which reduces the computational complexity [13]. Additionally, this method has a high payload capacity.

The first blockchain-like protocol was proposed by Cryptographer David Chaum in his 1982 dissertation "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups" [25]. However, it was not until 2008 that Bitcoin, the first application of blockchain, was proposed by Satoshi

Nakamoto, which started a new wave of blockchain technology. As the name suggests, a blockchain is a chain of blocks. Each block contains main data, a timestamp, and a pointer pointing to the previous block (parent block). These pointers are cryptography hashes [33]. Once the content of a block is tampered with, its hash also be changed, and subsequently, the pointer of the next block becomes invalid, and the entire blockchain is broken. Therefore, any modifications in the blockchain is recorded and reflected, which makes the blockchain tamper-evident. Based on this ability, in the proposed work, a verification and recovery mechanism is designed to improve the robustness to noises.

In this project, we propose a secure photo transmission mechanism based on steganography and blockchain. The secret photo, which can either be grayscale or color, is hidden from attackers by using the steganography technique. Then, secret pixels are embedded in a blockchain and become tamper-evident. Additionally, by using a suitable recovery mechanism, lost or modified data is recovered, which increases the robustness of image transmission.

1.1 Motivation

With the development of digital technology and the Internet, more and more people choose to transfer their photos over the network [4]. Many of these photos contain private or confidential information. However, security threats on the Internet are always present. Therefore, many cryptography-based schemes are proposed to improve the security of secret images by mapping them into random pixels. These methods might improve the secrecy of images to some extent, but still are vulnerable to data loss and malicious tampering during the transmission. Therefore, the suitable solution should also take the robustness to noises and attacks into consideration.

1.2 Contributions

Due to presence of malicious attackers over the network, the security of image transmission is always threatened. In recent years, many secure transmission mechanisms have been proposed. They are based on cryptography, steganography, and even deep learning methods. These schemes improve the security of transmission to a certain extent. They all focus on hiding secret pictures to prevent the intruder from obtaining information. However, the problem at the receiving end is often ignored. In network transmission, data loss often occurs. For the receiver, although the secrecy of the photo has not

been revealed, its integrity has been compromised. Hence, the decoded image may loss some vital information. To overcome this problem, we proposed a recovery mechanism to improve the integrity of the secret image.

We outline the overall contributions of this work below.

- Proposed a novel secure and robust photo transmission mechanism based on steganography and blockchain.
- Design of a novel recovery mechanism to improve the robustness of transmission.
- Implement an Android application for simulation of the proposed mechanism.
- Evolution of the performance of the proposed scheme through simulations.

1.3 Organization

The remainder of the paper is organized as follows: Section 2 introduces the related work in the area. The proposed mechanism is described in Section 3. Section 4 demonstrates the system setup and gives the results. Finally, conclusions and future work are given in Section 5.

2 Related Work

Due to the rapid development of network technology, a large number of photos are transmitted on the network every day. These photos usually contain confidential or private information. Meanwhile, the safety of these pictures is constantly under threat. To overcome it, many secure image transmission mechanisms were proposed. Most of these mechanisms were based on two techniques: image encryption and data hiding. This section briefly introduces and analyze the recent image security methods. A simple comparison between the proposed B-Spot and existing methods are also given.

2.1 Image Encryption

Image encryption is an effective way to protect secret images from unauthorized access. Basically, for encryption, a secret image is converted into an encrypted image by applying a certain encryption algorithm. Reversely, the original image can be decrypted from the cipher image by following the decryption algorithm. There are two major categories of encryption: Symmetric encryption and Asymmetric encryption. Some recent image encryption schemes are described below.

Bouslimi et al. [2] introduced a secure medical image scheme that combined a substitutive watermarking algorithm, the quantization index modulation, with an encryption technique, AES or RC4. They claimed that the decrypted image had low distortion and the embedding capacity was high. However, the proposed approach had high complexity and less robustness to image compression. A novel image encryption scheme based on the chaotic tent map was proposed by Li et al. [16]. They claimed that the scheme achieved better randomness properties and security levels. According to the results of the fail-safe analysis, it also had high efficiency. But this method didn't show its robustness to noises. Oravec et al. [20] introduced an image encryption algorithm based on chaotic maps. Properties of Arnold's cat map were employed for creating a relation between the used key and plaintext in form of image pixel amplitudes. As they claimed, the main advantage of the proposed method was its simplicity which enables fast processing speed. However, it had lower values of NPCR, a parameter used to measure the resistance to differential attacks, compared with other solutions. Singh et al. [26] proposed a new dynamic AES algorithm developed by key-dependent dynamic S-Box using dynamic irreducible polynomial and affine constant to avoid the risk of algebraic attacks. The results claimed the proposed method performed better in the coefficient analysis as compared to standard AES.

2.2 Image Steganography

Image steganography is a vital application of steganography, whose carrier used to hide the secret message in an image [10]. Image steganography methods can be classified to the spatial domain or transform domain based on the embedding domain. Payload capacity (bpp) and Imperceptibility (PSNR) are two common measures used to evaluate an image steganography method. Some recent image steganography methods are introduced as follows.

Muhammad et al. [17] proposed a secure image steganographic framework based on the stego key-directed adaptive least significant bit (SKA-LSB) substitution method and multi-level cryptography. In the proposed scheme, a two-level encryption algorithm (TLEA) and a multi-level encryption algorithm (MLEA) were used to encrypt the stego key and secret data respectively, and the encrypted information was then embedded in the cover image using an adaptive LSB substitution method. The proposed method kept a balance between security and imperceptibility. But, it had limited capacity (1bpp) and robustness. A new fuzzy-based adaptive LSB modification

method was proposed by Dadgostar and Afsari [7], which had better imperceptibility. In the proposed scheme, an interval-valued intuitionistic fuzzy edge detector was used to detect the edge areas. Then more secret bits were embedded into the edge regions than smooth regions which reduce the perceptible distortion. Experimental results showed that, it obtained high PSNR values and embedding capacity. But it had not been tested against different attacks. Moreover, the secret image was not reversible so it cannot be applied to images that require high fidelity. Kalita and Tuithung [14] proposed a spatial domain method by combining 8nPVD and LSB substitution. The cover image was partitioned into 3x3 non-overlapping pixel blocks in row-major order. By using a modified LSB algorithm, secret pixels were embedded in the center of the block. As the results showed, this method had a high payload capacity and it was simple to extract the secret image. However, it resulted in poor security and poor robustness against attacks. Saidi et al. [24] proposed a chaotic map and DCT based steganographic method. The proposed method embedded the secret bits in the least and the medium DCT coefficients of the cover photo according to the outputs of the PWLCM function. A quantitative study showed the proposed method met the requirement of imperceptibility and flexibility. However, the quality of the stego-image was not satisfied. It was also not robust against geometric and compression attacks.

After investigating some existing methods, it can be found that most of the existing secure image techniques can improve the security of images to a certain level. But they were not robust enough to data loss and malicious tampering during transmission. Therefore, in the proposed scheme B-Spot, a recovery mechanism is introduced to add extra robustness to noise. Before that, a 3-3-2 LSB method is used to hide the secret image. This method is simple to understand and implement so that it has low computational complexity. Moreover, this method has a high embedding capacity and keeps the fidelity of the cover image, as well.

3 B-Spot: The Proposed Mechanism

Based on steganography and blockchain, a novel secure and robust photo transmission mechanism is proposed. Figure 1 shows the overview of the proposed mechanism, that the proposed scheme mainly consists of a sender and a receiver. In the following sections, the specific steps within two components is described.

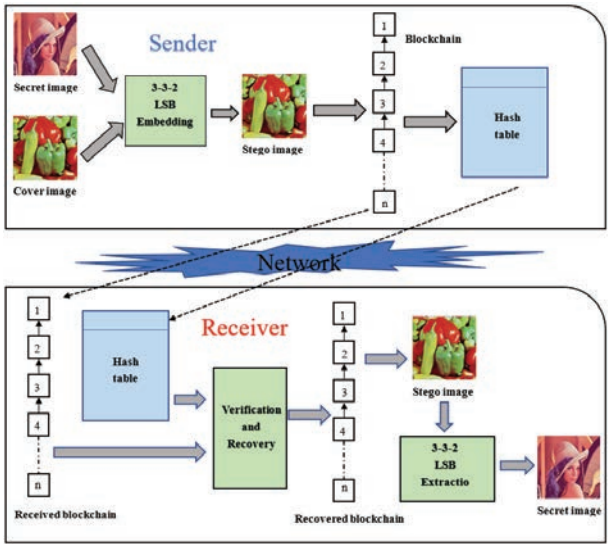

Figure 1 Overview of the proposed mechanism.

3.1 Sender

3.1.1 LSB embedding

Image steganography takes advantage of the limitations of the human eye. RGB images are usually used as cover images to hide secret data. A good image steganography algorithm needs to embed enough secret data while ensuring the fidelity of the cover image. The proposed mechanism uses 24 -bit RGB images as cover images to hide either grayscale or color secret photos. A 3-3-2 LSB method is used for color image steganography [23].

The 3-3-2 LSB method requires that 8-bit secret data is embedded into the Red, Green, and Blue channels of the secret image in the order of 3 bits, 3 bits, and 2 bits, respectively. Specifically, the first two 3 bits of secret data are concealed in the R and G channels, and the remaining 2 bits are hidden in the Blue channel. The reason for this is that human eyes are more sensitive to blue than red and green [8]. Besides, when our secret photo is a 24-bit RGB photo, it overall takes 3 pixels to hide all of the R, G, B channels of one secret pixel. Therefore, the cover image used by this mechanism must be at least three times the size of the secret photo. The detailed embedding process for both grayscale and color photos is described in Figure 2. Algorithm 1 shows the embedding process.

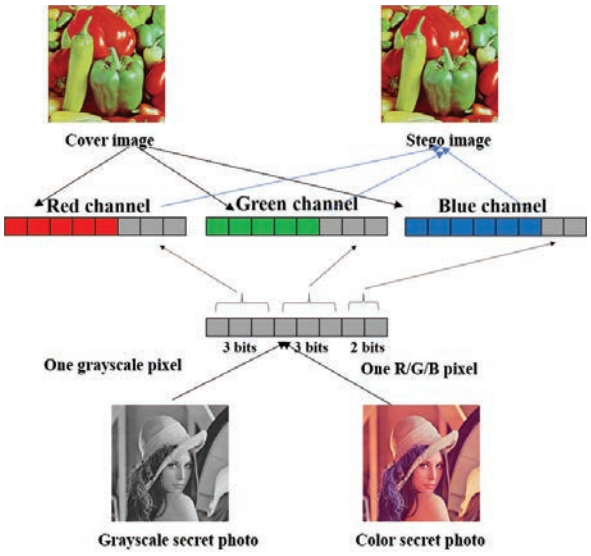

Figure 2 The process of 3-3-2 LSB embedding.

3.1.2 Blockchain building

Blockchain is a chain data structure composed of blocks. Each block has the same structure. Usually, it contains two parts: block header and block body. The block header consists of the timestamp of the block creation, the hash value of the previous block, and a nonce. The block body contains data. The hash value of each block takes its block header and block body as input and is calculated by a hashing function. Therefore, when the data of a block is modified, its hash value must also change. The hash value is also a link connecting two blocks. The link is destroyed when the block is changed. For many applications, the block body usually contains transactions. Additionally, the blockchain is deployed in a distributed network, and each node is a blockchain. But in the proposed mechanism, the blockchain is only regarded as a chain data structure. We use its feature of tampering-evident to build a robust data recovery mechanism.

As shown in Figure 3 the blockchain in the proposed mechanism is a single chain. Each block also contains a block header and a block body. The block header only contains the hash value of the previous block and the creation timestamp of this block. The data contained in the block body is the pixels of the stego-image. One block contains 6 sequential secret pixels. Adjacent blocks are all connected by the hash value of the parent block. The hash value is calculated by the hashing function of SHA-256 [19].

Algorithm 1 3-3-2 LSB embedding

Input:
$S \leftarrow$ Array of secret image pixels
$C \leftarrow$ Array of secret image pixels
Output:
$ST \leftarrow$ Array of stego image pixels

Begin
if S is a grayscale image **then**
  for $i = 1$ to $|S|$ **do**
    $c_r \leftarrow$ the red channel of $C(i)$
    $c_g \leftarrow$ the green channel of $C(i)$
    $c_b \leftarrow$ the blue channel of $C(i)$
    $p_r \leftarrow (c_r$ & 0xf8) | $(S(i) >>> 5)$
    $p_g \leftarrow (c_g$ & 0xf8) | $((S(i) >>> 1)$ & 0x07)
    $p_b \leftarrow (c_b$ & 0xfc) | $(S(i)$ & 0x03)
    Insert p_r, p_g, p_b to the R, G, B channel of $ST(i)$
  end for
else
  $j \leftarrow 0$
  for $i = 1$ to $|S|$ **do**
    for each $s \leftarrow$ one channel of $S(i)$ **do**
      $j \leftarrow j + 1$
      $c_r \leftarrow$ the red channel of $C(j)$
      $c_g \leftarrow$ the green channel of $C(j)$
      $c_b \leftarrow$ the blue channel of $C(j)$
      $p_r \leftarrow (c_r$ & 0xf8) | $(s >>> 5)$
      $p_g \leftarrow (c_g$ & 0xf8) | $((s >>> 2)$ & 0x07)
      $p_b \leftarrow (c_b$ & 0xfc) | $(s$ & 0x03)
      Insert p_r, p_g, p_b to the R,G,B channel of $ST(j)$
    end for
  end for
end if
return ST
end

It is important that once such a blockchain is built, any changes is recorded unless the entire blockchain is reconstructed. For example, in Figure 3, data 1 of block 2 has been tampered with, and the newly calculated hash value is different from the previously stored hash value in block 3. If you try to change the hash value of the parent block in block 3 to the new hash value, it also causes the hash value of block 3 to change. Therefore, you need to reconstruct the rest of the blocks to conceal one modification in block 2. This

Algorithm 2 Blockchain building

Input:
$ST \leftarrow$ Array of stego image pixels
Output:
$BC \leftarrow$ The blockchain containing all stego pixels

Begin
$p_h \leftarrow 0$
for $i = 1$ to $|ST|$ **do**
  $t \leftarrow$ the current time stamp
  for $j = 1$ to 6 **do**
    $d_j \leftarrow ST(i)$
    $i \leftarrow i + 1$
  end for
  $BC.append(p_h, t, d_1, d_2, d_3, d_4, d_5, d_6)$
  $p_h \leftarrow$ SHA-256$(p_h + t + d_1 + d_2 + d_3 + d_4 + d_5 + d_6)$
end for
return BC
end

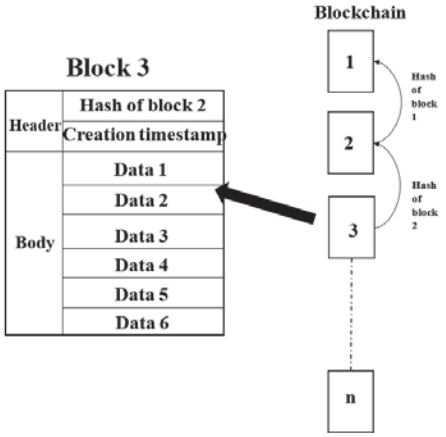

Figure 3 The structure of the blockchain.

is the feature of tampering-evident provided by the blockchain mechanism, which ensures that any changes to the block is recorded. This feature provides us with benefits for building a recovery mechanism. When transmitted in the network, if the data in the block has been maliciously tampered with or lost. We only need to check the connection between the blocks at the receiving end to verify the integrity of the data. Then the recovery mechanism is used to recover the damaged blocks.

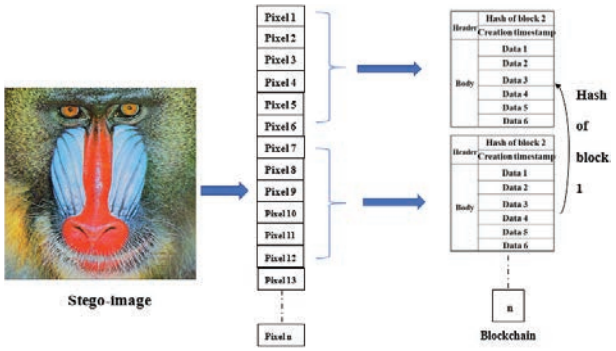

Figure 4 The process of blockchain building.

The construction process of the blockchain is shown in Figure 4. Firstly, take the stego-image and get the first six pixels. They are embedded into the block body as data 1 to data 6. Then the current timestamp is recorded. Since this is the first block, the hash value of the previous block is set to 0. Now, the first block is successfully built, which is called a genesis block. The hash value of the first block can be calculated by SHA-256, and it is stored in the block header of the second block. The above process is repeated until all pixels of the stego-image are embedded into the blockchain. Algorithm 2 demonstrates the above process in detail.

3.1.3 Hash table creation

To recover from the failure, we need extra redundancy. In the proposed mechanism, a hash table is designed to be the replication of the blockchain for recovering data loss and tampering. A hash table is a key-value data structure, with which we can search a certain value by a unique key at high speed. In our design, the hash table keep a copy of every content in the blockchain. Every entry in the hash table is equivalent to a block in the blockchain. The key of an entry is the index of the block in the same entry. So, a particular block can be searched by its index in the hash table. To create a hash table, we can simply scan the established blockchain from the first block to the last one. Upon every single block, firstly, an array length of 8 needs to create to contain the content of the block: 6 pixels, timestamp, and the hash value of the parent block. This array is the value of the entry. The key of the entry is the index of the current block. By using *put* $\langle key, value \rangle$ operation, one block is then transformed to an entry of the hash table. The above process should be repeated for all blocks in the blockchain. Figure 5 (a) demonstrates the process of creating a hash table. Algorithm 3 describes the process in detail.

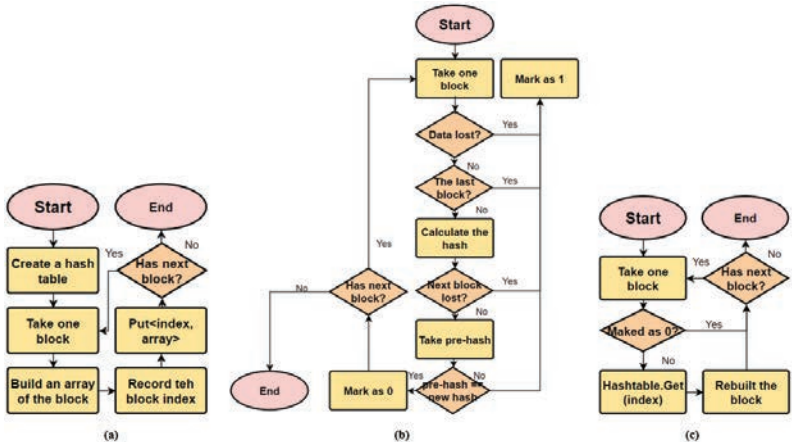

Figure 5 (a) Flow chart of the hash tabled creation (b) Flow chart of verification process (c) Flow chart of the recovery process.

Algorithm 3 Hash table creation

Input:
$BC \leftarrow$ The established blockchain
Output:
$HT \leftarrow$ The hash table as a copy of blockchain

Begin
for $i = 1$ to $|BC|$ **do**
  $h_p \leftarrow BC(i).h_p$
  $t \leftarrow BC(i).timestamp$
  $d_1 \leftarrow BC(i).data_1$
  $d_2 \leftarrow BC(i).data_2$
  $d_3 \leftarrow BC(i).data_3$
  $d_4 \leftarrow BC(i).data_4$
  $d_5 \leftarrow BC(i).data_5$
  $d_6 \leftarrow BC(i).data_6$
  $v \leftarrow (h_p, t, d_1, d_2, d_3, d_4, d_5, d_6)$
  $k \leftarrow i$
  $HT.put(k, v)$
end for
return HT
end

3.2 Receiver

On the receiver side, the integrity of the received blockchain is firstly verified. The process of verification utilizes blockchain's feature of tampering-evident

to check whether data loss or malicious tampering happened in a certain block during transmission. After that, a recovery process is executed to recover these modified blocks marked by the verification process. Then, the stego-image is reconstructed from the recovered blockchain. Finally, by following the LSB extraction process, the secret photo can be obtained. The details of each process is described as follows:

3.2.1 Verification

During the transmission, various attacks may occur, which compromise the data integrity of the blockchain. Although the lost data may not be understood by the attackers due to the image steganography. We still need the intact data to obtain the original secret image. However, some modifications to the data are not easily detectable. For example, the attacker may only change one least bit of the pixel, which cannot be perceived by human beings. But one LSB of the stego-image may be one MSB of the secret photo, which contains most information of a secret pixel. Therefore, before recovering the tampered data, the first step is to identify the data loss and data tampering in the blockchain. Due to the tampering-evident feature of the blockchain, any change in any single block is recorded. Based on this feature, a verification process is used to verify the integrity of the blockchain. The process is shown in Figure 5(b) Firstly, the hash value is recalculated for the current block header and block body using the same SHA-256 hashing function. Then the newly calculated hash value is compared with the original hash value stored in the next block. If these two hash values are equal, it suggests the data in the current block is not tampered with. Otherwise, this block should be marked. An array is maintained to record the result of verification. All elements are initialized as 0, but the position of any modification and loss is marked as 1. Also, if the next block is missing, the current one is marked as 1. As for the lost blocks, they are directly marked as 1. Algorithm 4 describes the process of verification.

3.2.2 Recovery

After the verification process, the positions of data loss and tampering have been marked. The next step is the recovery process. The hash table is the key to the recovery. It provides the redundancy to recover missing or modified data. Every block marked as 1, is rebuilt by referring to the hash table. Figure 5(c) demonstrates the general process in a flow chart. The array marking the data loss and data tampering is scanned. If the element is 1, we search the respective block in the hash table by using *get* $\langle key \rangle$, where *key* is the current index. Then we create a new block from the returned array,

Algorithm 4 Verification

Input:
$BC \leftarrow$ The received blockchain
Output:
$M \leftarrow$ The array marking whether data is tampered with and lost or not at the certain position, 1 for destroyed, 0 for intact

Begin
for $i = 1$ to $|BC|$ **do**
  if $BC(i) = NULL$ **then**
    $M(i) \leftarrow 1$
  else if $i = |BC|$ **then**
    $M(i) \leftarrow 1$
  else if $BC(i+1) = NULL$ **then**
    $M(i) \leftarrow 1$
  end if
  $h_p \leftarrow BC(i).h_p$
  $t \leftarrow BC(i).timestamp$
  $d_1 \leftarrow BC(i).data_1$
  $d_2 \leftarrow BC(i).data_2$
  $d_3 \leftarrow BC(i).data_3$
  $d_4 \leftarrow BC(i).data_4$
  $d_5 \leftarrow BC(i).data_5$
  $d_6 \leftarrow BC(i).data_6$
  $h_n \leftarrow$ SHA-256$(h_p + t + d_1 + d_2 + d_3 + d_4 + d_5 + d_6)$
  $h \leftarrow BC(i+1).p_h$
  if $h_n \neq h$ **then**
    $M(i) \leftarrow 1$
  else
    $M(i) \leftarrow 0$
  end if
end for
return M
end

which replaces the broken one in the blockchain. The recovery process is demonstrated in Algorithm 5.

3.2.3 Stego-image reconstruction

Once the blockchain is recovered, the pixels inside the blockchain are also recovered. The stego-image can be reconstructed now. The reconstruction process is the reversed process of building blockchain. Algorithm 6 describes the reconstruction process.

Algorithm 5 Recovery

Input:
$BC \leftarrow$ The received blockchain
$M \leftarrow$ The array marking whether data is tampered with and lost or not at the certain position, 1 for destroyed, 0 for intact
$HT \leftarrow$ The received hash table
Output:
$RBC \leftarrow$ The recovered blockchain

1: **Begin**
2: **for** $i = 1$ to BC **do**
3: **if** $M(i) = 1$ **then**
4: $RBC(i) \leftarrow HT.get(i)$
5: **else**
6: $RBC(i) \leftarrow BC(i)$
7: **end if**
8: **end for**
9: **return** RBC
10: **end**

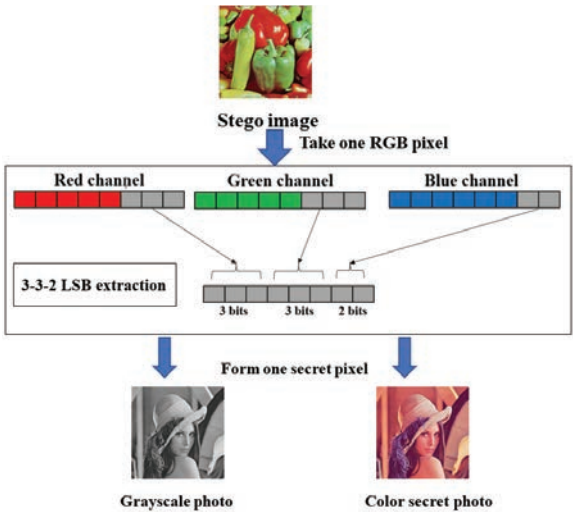

Figure 6 The process of 3-3-2 LSB extraction.

3.2.4 LSB extraction

After the stego-image is rebuilt from the recovered blockchain, the LSB extraction process should be performed to extract the secret photo from it. The detailed extraction process is depicted in Figure 6. The extraction process

Algorithm 6 Stego-image reconstruction

Input:
$RBC \leftarrow$ The recovered blockchain
Output:
$ST \leftarrow$ Array of stego-image pixles

Begin
$j \leftarrow 0$
for $i = 1$ to $|ST|$ **do**
    $d_1 \leftarrow RBC(i).data_1$
    $d_2 \leftarrow RBC(i).data_2$
    $d_3 \leftarrow RBC(i).data_3$
    $d_4 \leftarrow RBC(i).data_4$
    $d_5 \leftarrow RBC(i).data_5$
    $d_6 \leftarrow RBC(i).data_6$
    $j \leftarrow j + 1$
    $ST(j) \leftarrow d_1$
    $j \leftarrow j + 1$
    $ST(j) \leftarrow d_2$
    $j \leftarrow j + 1$
    $ST(j) \leftarrow d_3$
    $j \leftarrow j + 1$
    $ST(j) \leftarrow d_4$
    $j \leftarrow j + 1$
    $ST(j) \leftarrow d_5$
    $j \leftarrow j + 1$
    $ST(j) \leftarrow d_6$
end for
return ST
end

is straightforward since it is achieved in the reverse way of the embedding process. The specific steps of the 3-3-2 LSB extraction process are described in Algorithm 7.

4 Simulation Results and Analysis

4.1 Simulation Design

An Android application, named *B-Spot*, has been implemented to simulate the proposed blockchain and steganography based photo transmission mechanism. The application is tested running on an Android Studio AVD with 1 GB RAM and API 30. The simulation machine is a Windows 10 laptop with

Algorithm 7 3-3-2 LSB extraction

Input:
$ST \leftarrow$ Array of stego-image pixels
Output:
$S \leftarrow$ Array of secret image pixels

Begin
if S is a grayscale image **then**
  for $i = 1$ to $|S|$ **do**
    $st_r \leftarrow$ the red channel of $ST(i)$
    $st_g \leftarrow$ the green channel of $ST(i)$
    $st_b \leftarrow$ the blue channel of $ST(i)$
    $S(i) \leftarrow ((st_r \text{ \& 0xf7}) << 5) \mid ((st_g \text{ \& 0xf7}) << 2) \mid (st_b \text{ \& 0xf3})$
  end for
else
  $j \leftarrow 0$
  for $i = 1$ to $|S|$ **do**
    $j \leftarrow j + 1$
    $st_r \leftarrow$ the red channel of $ST(j)$
    $st_g \leftarrow$ the green channel of $ST(j)$
    $st_b \leftarrow$ the blue channel of $ST(j)$
    $s_r \leftarrow ((st_r \text{ \& 0xf7}) << 5) \mid ((st_g \text{ \& 0xf7}) << 2) \mid (st_b \text{ \& 0xf3})$
    $j \leftarrow j + 1$
    $st_r \leftarrow$ the red channel of $ST(j)$
    $st_g \leftarrow$ the green channel of $ST(j)$
    $st_b \leftarrow$ the blue channel of $ST(j)$
    $s_g \leftarrow ((st_r \text{ \& 0xf7}) << 5) \mid ((st_g \text{ \& 0xf7}) << 2) \mid (st_b \text{ \& 0xf3})$
    $j \leftarrow j + 1$
    $st_r \leftarrow$ the red channel of $ST(j)$
    $st_g \leftarrow$ the green channel of $ST(j)$
    $st_b \leftarrow$ the blue channel of $ST(j)$
    $s_b \leftarrow ((st_r \text{ \& 0xf7}) << 5) \mid ((st_g \text{ \& 0xf7}) << 2) \mid (st_b \text{ \& 0xf3})$
    Insert s_r, s_g, s_b to the R, G, B channel of $S(i)$
  end for
end if
return S
end

Inter(R) Core i5-7200U CPU and 8 GB RAM. Figure 7 displays the welcome UI, sender UI, and receiver UI of the application.

In the sender UI, users can firstly select secret and cover images from the storage by invoking a gallery app. Then they should tell the system the type of the secret image, grayscale or color image. By clicking the *Embed*

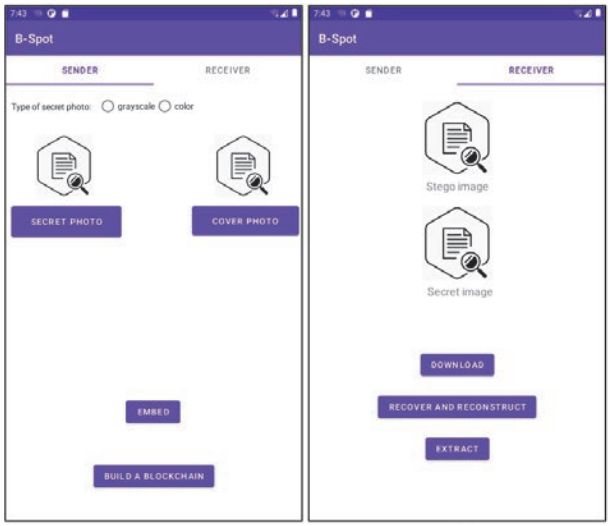

Figure 7 The UIs of B-Spot application, (a): the sender UI, (b): the receiver UI.

button, the secret photo is embedded into the cover image by following the 3-3-2 LSB method. Next, uses can build a blockchain from the stego-image by clicking the *Build a blockchain* button. Finally, the established blockchain can be sent to the server over the network by clicking the *Upload* button. As for the receivers, they need to firstly download the blockchain from the server by clicking the *Download* button. To obtain the stego-image, the *Recover and Reconstruct* button should be clicked, which perform the verification, recovery, and stego-image reconstruction sequentially. After that, the secret image can be extracted from the stego-image by clicking the *Extract* button.

The proposed mechanism has been tested on several images taken from the dataset of the University of Wisconsin-Madison [6]. In this paper, evaluation of the proposed scheme is demonstrated both for grayscale and color secret images, the secret images and cover images are shown in Figure 8.

4.2 Results and Analysis

4.2.1 Performance metrics

The Peak Signal to Noise Ratio (PSNR): This parameter is used to measure imperceptibility between images in dB. It is used to evaluate the performance of the 3-3-2 LSB method and the proposed recovery mechanism using Equation (1) [30]. The higher value of PSNR indicates more imperceptibility. The

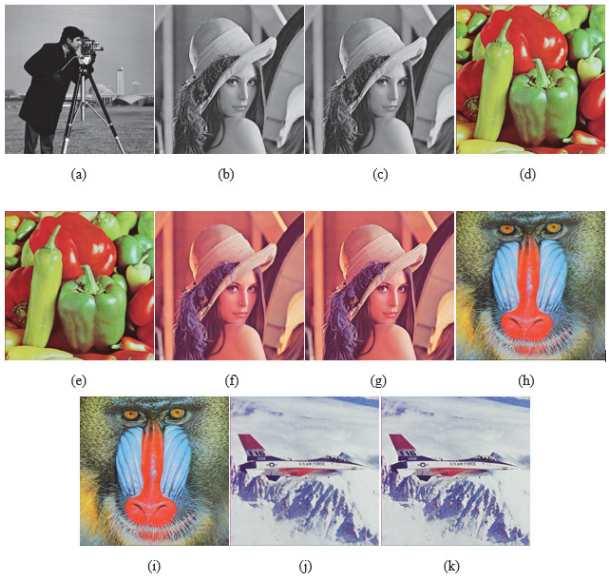

Figure 8 Secret images: (a) 256*256 grayscale Cameraman (b) 256*256 grayscale Lena (c) 512*512 grayscale Lena Cover images: (d) 256*256 color Peppers (e) 512*512 color Peppers (f) 256*256 color Lena (g) 512*512 color Lena (h) 256*256 color Baboon (i) 512*512 color Baboon (j) 256*256 color Airplane (k) 512*512 color Airplane.

PSNR is calculated based on Mean Square Error (MSE) calculated using Equation (2). The MSE is used to measure the error between images.

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right) \tag{1}$$

$$\begin{aligned} MSE_R &= \frac{1}{H \times W}\sum_{x=0}^{H-1}\sum_{y=0}^{W-1}(R(x,y) - R'(x,y))^2 \\ MSE_G &= \frac{1}{H \times W}\sum_{x=0}^{H-1}\sum_{y=0}^{W-1}(G(x,y) - G'(x,y))^2 \\ MSE_B &= \frac{1}{H \times W}\sum_{x=0}^{H-1}\sum_{y=0}^{W-1}(B(x,y) - B'(x,y))^2 \\ MSE &= \frac{MSE_R + MSE_G + MSE_B}{3} \end{aligned} \tag{2}$$

For color images, at first, the MSE of RBG channels should be calculated separately. Then, the average value of R, G, B MSEs should be calculated to represent the overall MSE for the color image. The R, G, B stands for the red, green, and blue channels of the original image. The R', G', B' are the red, green, and blue channels of the resulted image.

Structural Similarity (SSIM) is a parameter for measuring the similarity between two images [31]. It ranges from 0 to 1. A value closer to 1 indicates a higher similarity between the two images. In this paper, it is used as another metric to measure the performance of the proposed mechanism. SSIM can be calculated using Equation (3)

$$\begin{aligned} SSIM(X,Y) &= \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \\ SSIM(X_{rgb}, Y_{rgb}) &= \frac{1}{3}\sum_{c=1}^{3} SSIM(X_a, Y_a) \end{aligned} \tag{3}$$

where X denotes one image, Y denotes another image. μ_X and μ_Y are the mean of two images, σ_X^2 and σ_Y^2 are the variances of X and Y, respectively, σ_{XY} is the co-variance of X and Y. C_1 and C_2 are used for stability. For color images, we need to calculate the SSIMs for all RGB channels. The average value of 3 SSIMs is the overall SSIM for the color image.

4.2.2 Imperceptibility

Good imperceptibility makes it less possible for attackers to notice the existence of a secret image in the stego-image, which improves the security of the secret image during transmission. To evaluate the imperceptibility of the 3-3-2 LSB method, PSNR and SSIM are calculated for cover images and stego-images of different types. In Table 1, PSNR and SSIM of the 3-3-2 LSB embedding are recorded for secret images and cover images in Figure 8. It can be seen that PSNRs for all secret images are around 40. The SSIMs are also close to 1, which indicates the high similarity between cover images and stego-images. Figure 9 shows the extracted secret images of grayscale 256×256 Lena and corresponding stego-images, which all have high image quality. The imperfections cannot be detected by human eyes directly. Histogram analysis is a popular technique to show the distribution of image pixels. It can report small differences more accurately than human eyes. Figure 10 compares the histograms of original secret images and extracted secret images. According to these histograms, the extracted image

Table 1 The imperceptibility of 3-3-2 LSB technique measured in PSNR (dB) and SSIM

Secret image \ Cover image	256×256 color Lena		256×256 gray Lena		512×512 gray Lena	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
256×256 Peppers			40.41612	0.96618		
512×512 Peppers	41.37819	0.96319			40.68860	0.95844
256×256 Baboon			40.49087	0.99046		
512×512 Baboon	41.45946	0.98845			40.76306	0.98764
256×256 Peppers			40.45531	0.96565		
512×512 Peppers	41.42630	0.95914			40.70607	0.96013
256×256 Peppers			40.50390	0.97112		
512×512 Peppers	41.45317	0.96228			40.75407	0.96499

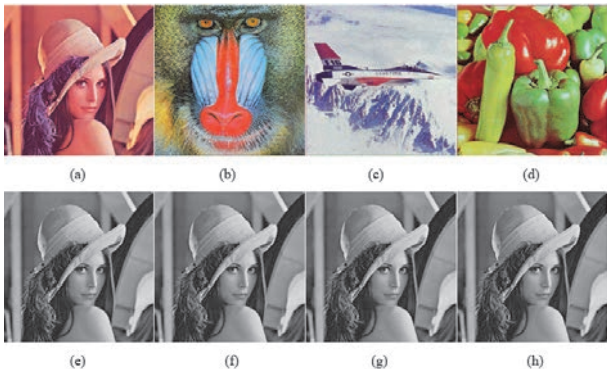

Figure 9 (a) (b) (c) (d): 256*256 Stego images (e) (f) (g) (h): Extracted 256*256 grayscale Lena secret images.

is the same as the original secret image. Therefore, the 3-3-2 LSB method has high imperceptibility and fidelity.

4.2.3 Embedding capacity

The embedding capacity is defined as the number of the secret bits embedded into one pixel of the cover image. A good steganography technique should achieve high embedding capacity while maintaining the high quality of stego-images. The embedding capacity can be calculated using Equation (4)

$$EC = \frac{Bits}{W \times H \times C}(bpp) \tag{4}$$

Where EC is embedding capacity, Bits is the maximum number of secret bits that can be hidden in the cover image, W is the width of the cover image, H is the height of the cover image, C represents the number of channels

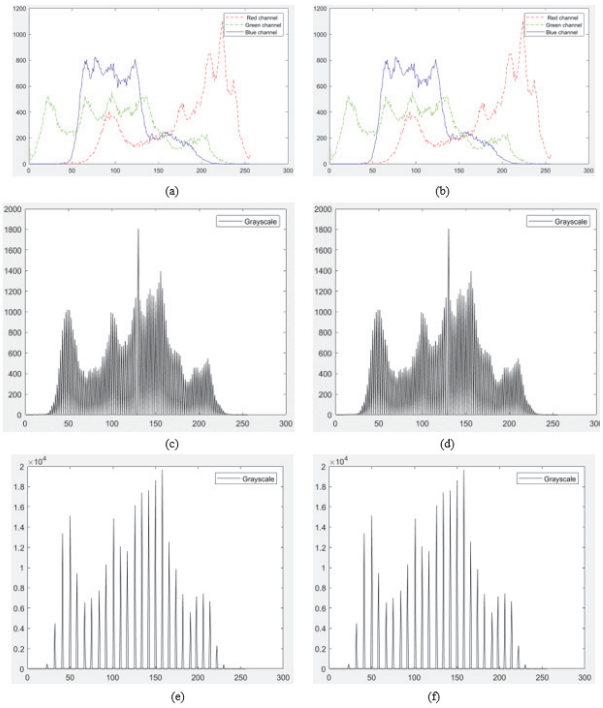

Figure 10 (a) original color Lena (b) extracted color Lena (c) original 256*256 grayscale Lena (d) extracted 256*256 grayscale Lena (e) original 512*512 grayscale Lena (f) extracted 512*512 grayscale Lena.

(3 for color image and 1 for grayscale image). EC is measured in bpp (bits per pixel).

For the 3-3-2 LSB method, the maximum number of secret bits is $768/(256 \times (3+3+2))$. The width and height of the cover image are 256. The cover image is an RGB image. Thus, the EC for 3-3-2 LSB is calculated as Equation (5)

$$EC = \frac{768}{256 \times 256 \times 3} \approx 2.67(bpp) \tag{5}$$

Therefore, the 3-3-2 LSB method has high embedding capacity as well as good imperceptibility.

4.2.4 Time consumption

The time consumption is also important for a image transmission mechanism, especially in a real-time application. In this paper, the time consumption of

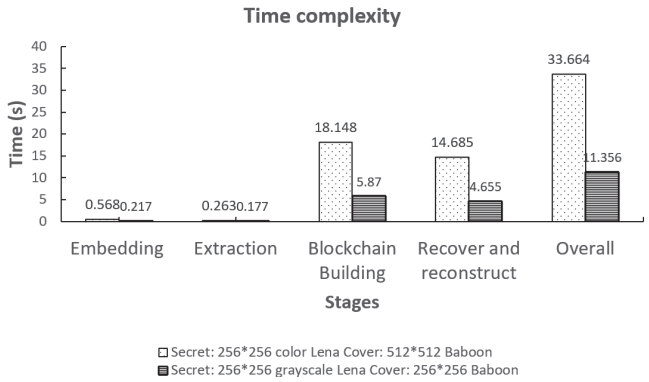

Figure 11 Histogram of the time complexity.

four stages: steganography embedding, steganography extraction, blockchain building, and blockchain recovery and reconstruction, as well as the total time of the whole process are measured. The 256×256 grayscale Lena image and 256×256 color Lena image are embedded into 256×256 color baboon image and 512×512 color baboon image, respectively. Table 2 records the time consumption of different stages in seconds. When the size of the cover image increases, the total time also increases. Figure 11 compares the time consumption in different stages using the histogram. It can be seen that the blockchain part takes more time than the steganography part. It is because of calculating hash and building new blocks are time-consuming. Thus, by increasing the number of pixels per block and working on a high-performance computer, the time complexity can be decreased.

4.2.5 Robustness

When the blockchain is transmitted over the network, it may be affected by various attacks. These attacks can be regarded as noises that result in data loss and malicious tampering. A good image transmission mechanism should have robustness to different levels of noise. In the proposed mechanism, we enhance the robustness by introducing a novel verification and recovery process. In this section, the salt-and-pepper noises (SPN) is added to the stego-images to simulate the data loss and malicious tampering. PSNR is used again to measure the similarity between the original secret image and extracted secret image. A grayscale 256×256 Lenna image and a color 256×256 Lena image are used as secret images. The 256×256 and 512×512 Airplane images are used as cover images. Figure 12 shows the noisy

Table 2 The time consumption of the proposed mechanism measured in second (s)

Steps / Secret / Cover	Embedding	Extraction	Blockchain Building	Recovery and Reconstruction	Overall
256×256 color Lena / 512×512 Baboon	0.568	0.263	18.148	14.685	33.664
256×256 gray Lena / 256×256 Baboon	0.217	0.177	5.87	4.655	11.356

Table 3 Comparison of robustness between proposed method with and without recovery process measured in PSNR (dB)

Secret Image / Density of SPN	256×256 gray Lena		512×512 gray Lena		256×256 color Lena	
	with recovery	without recovery	with recovery	without recovery	with recovery	without recovery
d = 0.001	Inf	37.18553	73.373042	36.141603	65.528553	36.137411
d = 0.005	48.765521	29.8114	46.733582	29.048124	45.271189	28.902695
d = 0.01	40.773668	26.355771	41.192013	26.123321	42.031027	25.971736
d = 0.05	27.916782	19.402752	27.559768	19.319688	27.643896	19.094679
d = 0.1	22.096406	16.221835	21.912222	16.27799	21.635757	15.965146

Table 4 Comparison of robustness between proposed method and other works measured in PSNR (dB)

Steps / Density of SPN / Secret image	256×256 color Lena		256×256 gray Lena		512×512 gray Lena	
	[32]	proposed	[5]	proposed	[29]	proposed
d = 0.002			46.9371	56.650178		
d = 0.005	28.16	45.271189	44.3189	48.765521		
d = 0.01	25.24	42.031027	36.105	40.773668		
d = 0.05	18.15	27.643896			22.2	27.9168
d = 0.08					20	23.7999
d = 0.1					19.1	22.0964

Figure 12 Salt and Peppers noises d = 0.01: (a) stego image (b) extracted secret image without recovery (c) extracted secret image after recovery Salt and Peppers noises d = 0.05: (d) stego image (e) extracted secret image without recovery (f) extracted secret image after recovery Salt and Peppers noises d = 0.1: (g) stego image (h) extracted secret image without recovery (i) extracted secret image after recovery.

stego-images, extracted images after recovery, and extracted images without recovery after being added different levels of SPN. By using the recovery process, the extracted images still have high fidelity after adding SNP of 0.1 density. Table 3 shows the PSNR of various extracted secret images and original secret images. It is obvious that the recovery mechanism improves the robustness greatly. The line graph in Figure 13 compares the robustness of the mechanism with and without the recovery process for the 256×256 color Lena secret image. The lost and modified data can be recovered by the recovery mechanism effectively. Compared with the single steganography technique, the blockchain technique helps improve the robustness of the mechanism significantly, especially when the density of noises is less than 0.01.

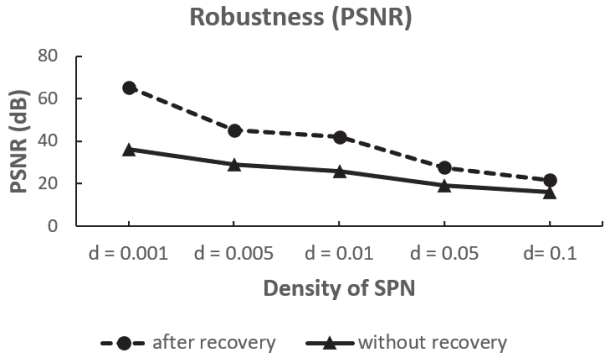

Figure 13 The PSNR of extracted secret images after adding different densities of SNP noises.

Table 5 Comparison of imperceptibility between 3-3-2 LSB and related works

	[15]	[34]	[28]	[21]	3-3-2 LSB
PSNR (dB)	33.2558	32.0671	33.0964	33.1779	40.3073

4.2.6 Comparison with related works

The robustness of the proposed mechanism is compared with existing secure image schemes [5,29,32]. In Table 4, a color airplane image size of 256×256 and another airplane image size of 512×512 are used as cover images to hide a grayscale Lena image size of 256×256, a grayscale Lena image size of 512×512, and a color Lena image size of 256×256. The salt and peppers noises, whose density ranges from 0.002 to 0.1, are added to the stego-images. According to the results, the proposed mechanism has stronger robustness compared with other schemes.

In Table 5, the imperceptibility of 3-3-2 LSB embedding is compared with other steganography methods using PSNR. In the comparison, the grayscale cameraman is embedded into a color Lena size of 256×256. The comparison results show that the PSNR in [15] is 33.2558 dB, [34] is 32.0671, [28] is 33.0964 dB, in [21], it is 33.1779 dB, and in the proposed scheme, the 3-3-2 LSB has PSNR of 40.307342 dB. From the comparison results, it can be concluded that our mechanism using the 3-3-2 LSB method has higher imperceptibility compared with some existing schemes.

In Table 6, the embedding capacity of 3-3-2 LSB embedding is compared with other LSB-related methods. The comparison results show that the embedding capacity in [18] is 1.5 bpp, [17] is 1 bpp [22] is 2 bpp and in [1], it is 2 bpp, and in the proposed scheme, the 3-3-2 LSB has EC of 2.67 bpp.

Table 6 Comparison of embedding capacity between 3-3-2 LSB and related works

	[18]	[17]	[22]	[1]	3-3-2 LSB
bpp	1.5	1	2	2	2.67

Therefore, our mechanism using 3-3-2 LSB embedding can also contain more secret data in the same cover image than other techniques.

5 Conclusion

In this paper, a blockchain and steganography based robust and secure photo transmission mechanism, named *B-Spot*, is proposed to provide security and robustness for photos being transmitted over the network. Blockchain technique is employed in the secure images scheme for the first time. In the proposed mechanism, either a grayscale photo or color photo can be embedded into the color cover image using the 3-3-2 LSB technique at the sender side. Then, the stego-image is broken down into pixels and inserted into blocks. All blocks connected by unique hash values form a blockchain. The blockchain is sent to the receiver together with a hash table. At the receiver side, after receiving the blockchain, a novel verification and recovery mechanism is executed to detect and recover the lost and destroyed data. Next, the recovered stego-image can be reconstructed from the blockchain. Finally, by following the extraction algorithm, the secret photo can be obtained. Due to the blockchain technique, this novel mechanism provides strong robustness to noises during the transmission compared with other existing secure image schemes. By using the 3-3-2 LSB technique, the proposed scheme also has good imperceptibility and high embedding capacity. The proposed method has been implemented as an Android application which demonstrates high performance. Moreover, *B-Spot* can also be combined with other existing security techniques such as encryption to add extra security and robustness.

References

[1] Shahzad Alam, Tanvir Ahmad, and Mohammad Najmud Doja. A novel edge based chaotic steganography method using neural network. In *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pages 467–475. Springer, 2017.

[2] Dalel Bouslimi, Gouenou Coatrieux, Michel Cozic, and Christian Roux. A joint encryption/watermarking system for verifying the reliability

of medical images. *IEEE transactions on information technology in biomedicine*, 16(5):891–899, 2012.

[3] Sammi Caramela. 8 tech security tips for creating a safe home office.

[4] Xiuli Chai, Zhihua Gan, Yiran Chen, and Yushu Zhang. A visually secure image encryption scheme based on compressive sensing. *Signal Processing*, 134:35–51, 2017.

[5] Xiuli Chai, Zhihua Gan, Kang Yang, Yiran Chen, and Xianxing Liu. An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and dna sequence operations. *Signal Processing: Image Communication*, 52:6–19, 2017.

[6] University of Wisconsin Computer Aided Engineering. Test images.

[7] H Dadgostar and Fatemeh Afsari. Image steganography based on interval-valued intuitionistic fuzzy edge detection and modified lsb. *Journal of information security and applications*, 30:94–104, 2016.

[8] Kousik Dasgupta, JK Mandal, and Paramartha Dutta. Hash based least significant bit technique for video steganography (hlsb). *International Journal of Security, Privacy and Trust Management (IJSPTM)*, 1(2): 1–11, 2012.

[9] Mehdi Hussain and Mureed Hussain. A survey of image steganography techniques. 2013.

[10] Mehdi Hussain, Ainuddin Wahid Abdul Wahab, Yamani Idna Bin Idris, Anthony TS Ho, and Ki-Hyun Jung. Image steganography in spatial domain: A survey. *Signal Processing: Image Communication*, 65:46–66, 2018.

[11] Joseph Johnson. U.s. data breaches and exposed records 2020 | statista.

[12] Neil F Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, 1998.

[13] Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, and Brendan Halloran. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing*, 335: 299–326, 2019.

[14] Manashee Kalita and Themrichon Tuithung. A novel steganographic method using 8-neighboring pvd (8npvd) and lsb substitution. In *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–5. IEEE, 2016.

[15] Vijay Kumar and Dinesh Kumar. Performance evaluation of modified color image steganography using discrete wavelet transform. *Journal of Intelligent Systems*, 28(5):749–758, 2019.

[16] Chunhu Li, Guangchun Luo, Ke Qin, and Chunbao Li. An image encryption scheme based on chaotic tent map. *Nonlinear Dynamics*, 87(1):127–133, 2017.

[17] Khan Muhammad, Jamil Ahmad, Naeem Ur Rehman, Zahoor Jan, and Muhammad Sajjad. Cisska-lsb: color image steganography using stego key-directed adaptive lsb substitution method. *Multimedia Tools and Applications*, 76(6):8597–8626, 2017.

[18] Tuan Duc Nguyen, Somjit Arch-Int, and Ngamnij Arch-Int. An adaptive multi bit-plane image steganography using block data-hiding. *Multimedia tools and applications*, 75(14):8319–8345, 2016.

[19] NIST. Nist brief comments on recent cryptanalytic attacks on secure hashing functions and continued security provided by sha-1, 2004.

[20] Jakub Oravec, Jan Turan, Lubos Ovseník, Tomas Ivaniga, David Solus, and Michal Marton. Asymmetric image encryption approach with plaintext-related diffusion. *Radioengineering*, 27(1):281–288, 2018.

[21] Shiv Prasad and Arup Kumar Pal. An rgb colour image steganography scheme using overlapping block-based pixel-value differencing. *Royal Society open science*, 4(4):161066, 2017.

[22] Sujarani Rajendran and Manivannan Doraipandian. Chaotic map based random image steganography using lsb technique. *IJ Network Security*, 19(4):593–598, 2017.

[23] Swati Sahute, Priyankaand Waghamare, Supriya Patil, and Ashwini Diwate. Secure messaging using image stegnography. *International Journal of Modern Trends in Engineering and Research*, 2(3):598–608, 2015.

[24] Marwa Saidi, Houcemeddine Hermassi, Rhouma Rhouma, and Safya Belghith. A new adaptive image steganography scheme based on dct and chaotic map. *Multimedia Tools and Applications*, 76(11):13493–13510, 2017.

[25] Alan T Sherman, Farid Javani, Haibin Zhang, and Enis Golaszewski. On the origins and variations of blockchain technologies. *IEEE Security & Privacy*, 17(1):72–77, 2019.

[26] Amandeep Singh, Praveen Agarwal, and Mehar Chand. Image encryption and analysis using dynamic aes. In *2019 5th International Conference on Optimization and Applications (ICOA)*, pages 1–6. IEEE, 2019.

[27] Ali Soleymani, Zulkarnain Md Ali, and Md Jan Nordin. A survey on principal aspects of secure image transmission. In *Proceedings of World*

Academy of Science, Engineering and Technology, volume 66. World Academy of Science, Engineering and Technology, 2012.

[28] Gandharba Swain. Adaptive pixel value differencing steganography using both vertical and horizontal edges. *Multimedia Tools and Applications*, 75(21):13541–13556, 2016.

[29] Zhenjun Tang, Xianquan Zhang, and Weiwei Lan. Efficient image encryption with block shuffling and chaotic map. *Multimedia tools and applications*, 74(15):5429–5448, 2015.

[30] Christian J Van den Branden Lambrecht and Olivier Verscheure. Perceptual quality measure using a spatiotemporal model of the human visual system. In *Digital Video Compression: Algorithms and Technologies 1996*, volume 2668, pages 450–461. International Society for Optics and Photonics, 1996.

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[32] Xiangjun Wu, Chenxi Bai, and Haibin Kan. A new color image cryptosystem via hyperchaos synchronization. *Communications in Nonlinear Science and Numerical Simulation*, 19(6):1884–1897, 2014.

[33] Karl Wüst and Arthur Gervais. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54. IEEE, 2018.

[34] Ching-Yu Yang and Wen-Fong Wang. Block-based colour image steganography using smart pixel-adjustment. In *Genetic and Evolutionary Computing*, pages 145–154. Springer, 2015.

Biographies

Dongchi Li is an MSc Computing (Software Engineering) student at Imperial College London. He is currently working on his Master Individual Project of Java Migration of MATLAB Line Library. He received a BSc Computer Science degree with first class from University of Nottingham, Ningbo, China in 2021. He is interested in software engineering, blockchain and distributed systems.

Pushpendu Kar received all this Bachelor, Master, and PhD in Computer Science and Engineering. He is currently working as an Assistant Professor with the School of Computer Science, University of Nottingham, Ningbo, China. Prior to this, he was a Research Fellow with the Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Norway. He was also a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, and the Energy Research Institute, Nanyang Technological University, Singapore. His research interests include mobile ad hoc networks, wireless sensor networks, the Internet of Things, and content-centric networking. He was a

recipient of the Erasmus Mundus Postdoctoral Fellowship of the European Commission, the ERCIM Alain Bensoussan Fellowship of the European Union, and the SERB OPD Fellowship of the Department of Science and Technology, Government of India. He received the 2020 IEEE Systems Journal Best Paper Award.