OXFORD

The Computer Journal, 2025, **68**, 1967–1979
https://doi.org/10.1093/comjnl/bxaf088
Advance access publication date 20 July 2025
Original Article

BlkInfoM: versatile blockchain-based mapping mechanism for secure information transmission

Yuanjing Luo [1,*], Xichen Tan [2], Jiaohua Qin [3], Zhiping Cai [2]

[1]College of Computer and Mathematics, Central South University of Forestry and Technology, No. 498 Shaoshan South Road, Tianxin District, Changsha, Hunan Province, 410004, China
[2]College of Computer, National University of Defense Technology, No. 109 Deya Road, Changsha, Hunan Province, 410073, China
[3]Hunan University, No. 1 Lushan South Road, Yuelu District, Changsha, Hunan Province, 410082, China
[*]Corresponding author. College of Computer and Mathematics, Central South University of Forestry and Technology, No. 498 Shaoshan South Road, Tianxin District, Changsha, Hunan Province, 410004, China. E-mail: luoyuanjing@csuft.edu.cn

Abstract

Information mapping is a widely adopted strategy in information hiding, leveraging the inherent features of carriers to convey hidden information without altering the carrier itself, thus maintaining integrity and avoiding detection. However, current mapping-based techniques face significant challenges, including potential data loss during transmission, limited capacity of carriers like images, and reliance on costly third-party storage solutions. To address these limitations, we introduce BlkInfoM, an innovative algorithm that utilizes blockchain's decentralized, immutable, and traceable properties as a novel data source for secure information hiding. BlkInfoM leverages blockchain transaction data, such as Merkle hash values, timestamps, and locations, in combination with a reversible ASCII-based binary encoding to enable precise information-to-block matching. Experimental results demonstrate that BlkInfoM not only improves the success rate and efficiency of information mapping compared to traditional methods but also reduces operational costs by eliminating the need for third-party storage. This work highlights the potential of blockchain technology to revolutionize information hiding, offering enhanced security, scalability, and cost-effectiveness.

1. INTRODUCTION

The rapid advancement of information technology has enabled remote data storage and sharing, often involving sensitive content not meant for public access [1]. Information hiding techniques play a crucial role in ensuring data security and privacy by embedding secrets within carriers such as text [2], images [3, 4], or video [5] using cryptographic methods [6]. However, these techniques typically modify carrier features, potentially causing detectable statistical anomalies that expose the hidden data to steganalysis [7, 8].

To fundamentally defend against steganalysis detection, Zhou *et al.* [9] proposed an information hiding strategy in May 2014, named that "coverless" steganography, which maps information onto the carrier's features without altering them. This approach preserves the integrity of the carrier and avoids detection caused by feature modifications. Since its inception, the coverless approach has rapidly evolved and found extensive application in the field of computer vision. Images, due to their accessibility and diversity, have become the most widely used mapping carriers [10, 11]. The focus of research is on identifying and utilizing image features to create efficient mapping rules [12–23]. While these methods have demonstrated good capacity and security performance, challenges remain: (i) the inherent instability of carriers like text, images, and videos risks data alteration or loss during transmission, potentially leading to inaccurate information retrieval; moreover, (ii) these carriers have limited features for mapping, reducing algorithmic capacity. This raises the question, "*Can we bypass the shortcomings of traditional carriers to design a new mapping architecture?*"

Blockchain technology offers a promising new approach for secure, decentralized information handling [24, 25], with its attributes of immutability, traceability, and permanent storage proving valuable in secret communication research. Recently, blockchain has been further applied to confidential communication through smart contracts and cryptographic methods. However, challenges remain, including (ii) privacy and cost concerns associated with forged Bitcoin addresses and large file storage [26]. Despite these challenges, blockchain's vast, immutable transaction data, particularly through Merkle roots generated by Merkle trees, ensures data integrity within networks like Bitcoin [24, 27, 28]. *These rich, reliable data sources provide ample opportunities for innovative information-hiding designs*.

To address these challenges, we propose BlkInfoM, a mapping mechanism based on blockchain transaction data. Our method downloads large volumes of block data and binary encodes their Merkle hash values. Using a reversible ASCII-based algorithm, plaintext information is converted into binary sequences and mapped to Merkle hash values, thus obtaining timestamps and data locations essential for precise information retrieval. To enhance mapping success and efficiency, the binary-encoded plaintext undergoes segmentation and pre-processing, creating an index structure containing "sequence-timestamp-location" pairs is constructed. For coverless steganography (see Fig. 1a), our approach requires only the transmission of keys, with the

Received: May 15, 2024. **Revised:** May 21, 2025. **Accepted:** June 14, 2025

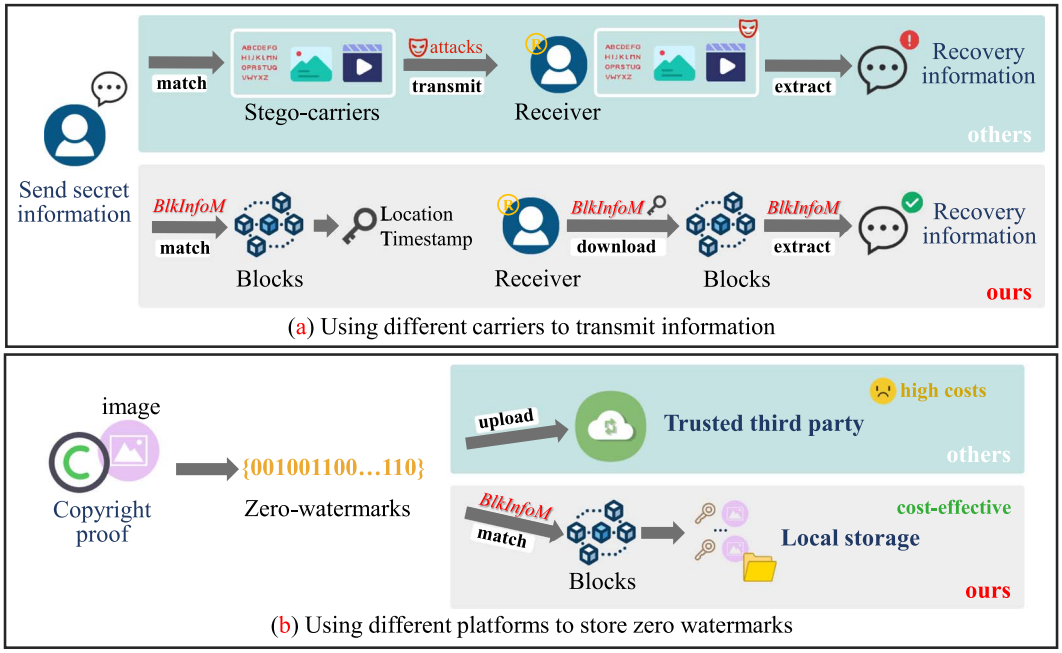

Figure 1. The difference between our BlkInfoM-based approach and others.

mapping carriers being immutable transaction data stored on the blockchain, fundamentally resisting steganalysis tools and preventing potential malicious attacks, thereby ensuring the security of covert communication (for Challenge 1). Benefiting from the richness of blockchain data, our scheme partially overcomes the limited capacity of existing coverless image/text/video algorithms (for Challenge 2). Additionally, leveraging existing block data enhances concealment and avoids the cost burden of uploading new data to the blockchain (for Challenge 3). By utilizing existing block data instead of generating new transactions, our method not only enhances concealment but also avoids the cost associated with data uploading to the blockchain (for Challenge 3).

This cost-effective solution also mitigates challenges faced by zero-watermarking algorithms, which often involve high costs and low efficiency due to third-party copyright data uploads. By creating an economical and reliable platform for local copyright proof storage and retrieval, BlkInfoM effectively addresses these issues (see Fig. 1b). In summary, this scheme offers the following contributions:

- We introduce BlkInfoM, which utilizes immutable blockchain transaction data for secure and covert communication, marking the first application of blockchain in this domain.
- The framework supports both coverless steganography and zero-watermarking, enhancing data security, privacy, and storage efficiency while minimizing costs.
- BlkInfoM improves success rates, increases embedding capacity, and eliminates carrier instability by leveraging blockchain's inherent properties.

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 introduces the proposed BlkInfoM, including its overview and essential matching processes. Section 4 evaluates BlkInfoM's performance under some basic setup. Section 5 further demonstrates the effectiveness of BlkInfoM in applications. Section 6 concludes this study.

2. RELATED WORKS

2.1. Mapping-based information hiding

Information hiding embeds secret information into carriers (e.g. images, audio, video, or text) to conceal it during transmission and storage without attracting third-party attention. It is widely used in covert communication, digital copyright protection, and data integrity verification. To overcome limitations of traditional methods, such as detection risk and carrier distortion, mapping-based techniques have been introduced. These approaches, which avoid altering the carrier, include coverless steganography and zero-watermarking, representing two key innovations in the field of information hiding.

2.1.1. *Coverless steganography*

As a critical branch of information hiding, steganography facilitates clandestine communication by embedding secret information into a carrier. However, embedding secret information often alters the carrier's features, leading to statistical anomalies and making the stego-carrier detectable by steganalysis tools [8]. To *fundamentally defend against steganalysis detection*, Zhou *et al.* [9] introduced the concept of "coverless" steganography in May 2014. Unlike traditional steganography, coverless steganography emphasizes directly mapping secret information to the carrier's inherent features, without altering them. This method has since advanced and is widely used in computer vision.

Images are frequently used as carriers in coverless steganography, with research focused on identifying and applying their distinctive features to create efficient mapping rules. For example, Zhang *et al.* [10] proposed a novel coverless image steganography algorithm based on discrete cosine transform and latent Dirichlet allocation topic classification. Luo *et al.* [12] introduced DenseNet to extract images' high-level features to establish the mapping relationship. Based on this, Luo *et al.* [13] further selected images' robust areas for feature mapping to against counter geometric attacks, and proposed a coverless image steganography method based on multi-object recognition to improve the robustness against geometric attacks [14]. Liu *et al.* [15] used DenseNet for feature extraction and Discrete Wavelet Transform (DWT) for sequence mapping, selecting suitable carrier images based on feature sequences. They also used camouflage images as carriers, correlating them with CNN features for secure transmission [16]. Furthermore, the Generative Adversarial Networks (GANs) introduced by Goodfellow *et al.* in 2014 [29] have been

extensively applied to coverless image steganography. By mapping secret information into GANs' noise vectors, it's possible to produce natural steganographic images embedding the secret information without any modifications [30]. Chen *et al.* [31] used a traditional CIS method to select a natural image for the initial secret data, mapped the remaining data to face attributes, and utilized StarGAN to generate a quality stego image from this relationship. Li *et al.* [32] encrypted and decrypted secret images across domains using two generative models, enhancing image quality and ensuring secure extraction. Peng *et al.* [33] proposed a novel image steganography based on the generator of GAN and gradient descent approximation to resolve the problem of the irreversibility in some common neural networks. Beyond images, text and video are also commonly used as mapping carriers in coverless steganography. Text, however, offers limited opportunities for modifications, leading to lower hidden data insertion rates and a higher likelihood of semantic anomalies after alterations [34–36]. Video mapping rules closely mirror those of image-based coverless steganography, with significant strides made toward improving robustness [37]. However, *carrier transmission, whether text, images, or videos, remains unstable, failing to fully resist all attacks. This results in a risk of carriers being altered or lost during transit, causing receivers to extract incomplete or incorrect information*.

2.1.2. *Zero-watermarking*

As another crucial branch of information hiding, watermarking is key in image copyright, addressing data protection and ownership verification issues [38, 39]. Traditional methods struggle to balance imperceptibility and robustness. In 2003, Wen *et al.* [40] introduced the concept of "zero-watermarking," utilizing high-order features of images to construct watermark information without changing the original image data, *achieving perfect imperceptibility and starting a new trend in watermark research*. Later research improved zero-watermarking's robustness by optimizing feature selection and construction. For example, Chen *et al.* [17] enhanced zero-watermark robustness with block segmentation and wavelet techniques. Dong *et al.* [18] countered affine transformations using image normalization. Chang *et al.* [11] combined neural networks and XOR operations with chaos sequences for stronger watermarks. Lai *et al.* [19] introduced digital signatures and timestamps for security. Yang *et al.* [20] applied normalization to improve resistance against attacks. Wang *et al.* [21] used low-frequency coefficients for feature-based zero-watermark creation. Yang *et al.* [22] developed a Zernike-Discrete Cosine Transform (DCT) algorithm for robust watermarking. Ren *et al.* [23] optimized spatial relationships with topological methods for clearer data mapping. Experimental results demonstrated these algorithms' resilience to geometric attacks.

Despite zero-watermarking's ability to overcome data loss issues, *it relies on a thirdparty for copyright information storage and arbitration, which can be costly to implement*. Blockchain's decentralization and smart contracts offer solutions, enhancing copyright protection and zero-watermarking reliability through blockchain-based registrations. Ren *et al.* [41] stored watermark and copyright data on the blockchain post-XOR. Wu *et al.* [42] developed a zero-watermarking algorithm using nonsubsampled contourlet transform (NSCT)-Singular Value Decomposition (SVD) and Arnold transform for more stable video features. Xu *et al.* [43] achieved robust zero-watermark images with K–L and NSCT transforms, stored via a blockchain system using IPFS and Hyperledger Fabric. Wang *et al.* [44] tackled blockchain data expansion issues with the IPFS (InterPlanetary File System). However, *challenges remain with high costs and low efficiency in using public blockchains like Ethereum*.

2.2. Blockchain

Blockchain, introduced by Satoshi Nakamoto in 2008, is fundamentally a multi-layered framework that includes data, network, consensus, execution, contract, and application layers [24]. At its core, blockchain operates as a distributed ledger that links data blocks through cryptographic algorithms, hash functions, and consensus mechanisms, ensuring tamper-resistance, transparency, and permanences [25, 45–47]. The framework's structure is key to its robustness: blocks are linked sequentially via timestamps and each contains the previous block's hash (*hashPrevBlock*), which reinforces the integrity of the chain. As shown in Fig. 2a, each block in this framework is composed of essential components, including the *Nonce*, *Bits* for proof of work, and a *Merkle Root*, a hash generated through the *Merkle Tree* structure. The *Merkle Tree* serves to condense all transactions in a block into a single hash, facilitating efficient and secure verification. Figure 2b illustrates this process: for instance, if a block contains transactions $\{TX_A, TX_B, TX_C, TX_D\}$, the *Merkle Tree* organizes them into leaf nodes storing their hash values. Non-leaf nodes then combine these hashes to create a single *Merkle Root*, summarizing all transactions within the block. This framework not only ensures data integrity but also supports efficient transaction verification, making it scalable for handling large volumes of data.

2.2.1. *covert communication on blockchain*

Blockchain's decentralized, immutable, and transparent properties have spurred research into its potential for covert communication. While blockchain provides a secure and tamper-resistant platform, leveraging it for covert communication requires embedding hidden data within transactions or smart contracts without affecting the system's core functionalities. Several studies have explored different methods for achieving this. Tschorsch and Scheuermann [48] conducted a comprehensive survey on decentralized digital currencies, highlighting the potential use of Bitcoin's transaction structure for embedding covert information. Specifically, fields such as OP_RETURN, which allow for the insertion of arbitrary data, have been investigated for hiding small amounts of information. However, as noted by Vasek and Moore [49], such methods come with the drawback of high detectability, as the use of OP_RETURN is not common in typical transactions and could be flagged for further analysis. In contrast, Zhang *et al.* [50] proposed using Ethereum smart contracts as a medium for covert communication. By embedding secret data within the execution process of smart contracts, their method conceals information in the contract's logic and execution results, making detection more difficult compared to direct transaction data manipulation. The flexibility of smart contracts allows for dynamic, decentralized covert channels that can operate without altering transaction patterns, thus offering a more sophisticated approach to hidden communication. In the latest study, Zhang *et al.* [51] developed a covert comms method for public blockchains using Shamir's threshold and STC mapping. It splits the master key into sub-keys via Shamir's scheme, shares them through blockchain transactions, and embeds secrets in a balanced transaction amount mapping, publishing them via transactions for covert communication.

Despite these advancements, challenges remain in using blockchain for covert communication. On-chain data storage

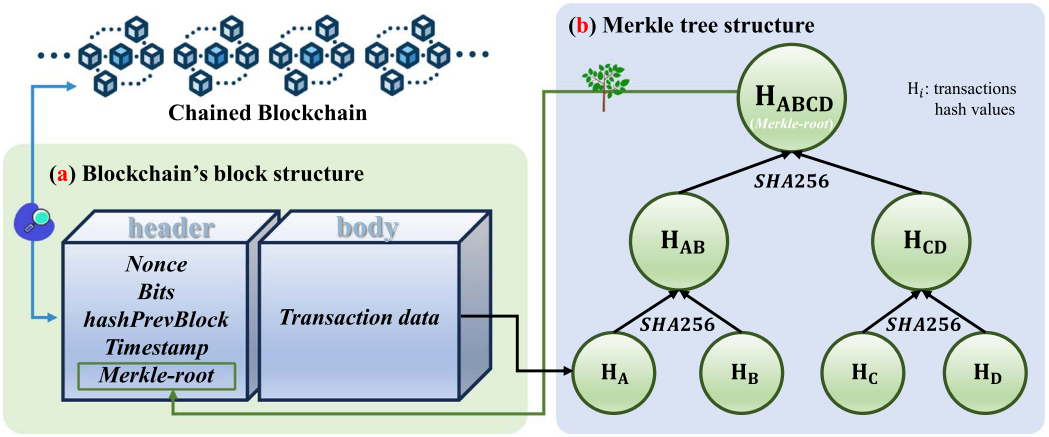

Figure 2. A detailed presentation of the block structure in Blockchain.

is costly, particularly in public blockchains like Ethereum. Secondly, the real-time nature of transaction verification can introduce delays, which may impact the effectiveness of time-sensitive communications. Additionally, the transparent nature of blockchain makes it susceptible to forensic analysis. While Heilman *et al.*[52] explored privacy-preserving techniques like Zerocoin and Tumblebit to enhance transaction anonymity, these systems were not specifically designed for covert communication. Embedding large amounts of covert data without detection remains challenging, as transaction patterns can still reveal anomalies [26].

3. BLKINFOM META-ARCHITECTURE

3.1. Overview

Our primary target is to explore a secure and reliable information match mechanism. Inspired by the robust capabilities of blockchain technology, we tightly integrate information with immutable transaction data on the blockchain, Merkle hash values. Given that data inherently exists in binary form, it necessitates the binary conversion of Merkle hash values. Moreover, to facilitate efficient matching from information to data, it is imperative to construct a dependable mapping dictionary is imperative. Figure 3 demonstrates the overview of the proposed BlkInfoM, highlighting its core components: on one hand, the meticulous processing of hash values to ensure the credible binary representation; on the other hand, the construction of a mapping dictionary, which is crucial for bridging information with blockchain data.

3.1.1. *Merkle hash processing*

Considering the uniform length of each Merkle hash value, we adopt a standardized binary conversion process. In our strategy, for each block, we initially extract its Merkle root to obtain a 64-bit Merkle hash sequence, comprised of digits and lowercase letters. We then treat this sequence as an 8 × 8 matrix, converting each character into its respective ASCII value and conducting a zigzag scan to construct an array $A = \{ASCII_1, ASCII_2, ..., ASCII_{64}\}$, which is further processed through:

$$\begin{cases} Q_a = 1, ASCII_i \le ASCII_{i+1} \\ Q_a = 0, ASCII_i > ASCII_{i+1} \end{cases}, 1 \le i \le 63. \quad (1)$$

Finally, each Merkle hash is transformed into a binary sequence capable of representing up to 63-bit binary sequence $Q = \{Q_1, Q_2, ..., Q_{63}\}$.

3.1.2. *Index construction*

Matching information with blockchain data presents some challenges, especially considering the variable lengths of information. Finding corresponding blockchain data for longer pieces of information may not be easy; moreover, conducting information retrieval within an irregular database is time-consuming. To address these issues, we have adopted a strategy of segmenting information and constructing an index structure to enhance the accuracy and efficiency of the match. In our approach, all information is divided into several 8-bit length segments for data matching. For effective index construction, we have included all possible 8-bit binary sequences as search entries, linking them to specific block lists. Each list meticulously records the block's corresponding timestamp and location information (see Fig. 3). The timestamp serves as a unique identifier for the block, enabling the precise location of the specific block, while the location information indicates the starting point of the binary sequence within the block's transaction information. Through this carefully designed index structure, we ensure a high success rate and efficiency in the accurate matching of information with data.

3.2. Information-to-block matching

In this section, we provide a detailed presentation of the process for matching information with blocks, which encompasses both the hiding of information within blocks and the extraction of information from blocks. Note that we assume all information to be processed is in plaintext form; thus, it necessitates reversible encryption. The specifics of this operation and methodology will be elaborated upon in the following text. Table 1 summarizes the notations in this paper.

3.2.1. *Information-block hiding*

To match the corresponding encrypted blocks, plaintext information must first be converted into binary sequences and segmented (we segment the information into segments of 8-bit each in this paper). Given the variability in the length of secret information and its inability to always be divisible by 8, we append several zeros to the end of the binary sequence to ensure uniformity in the length of all converted sequences. We provide a

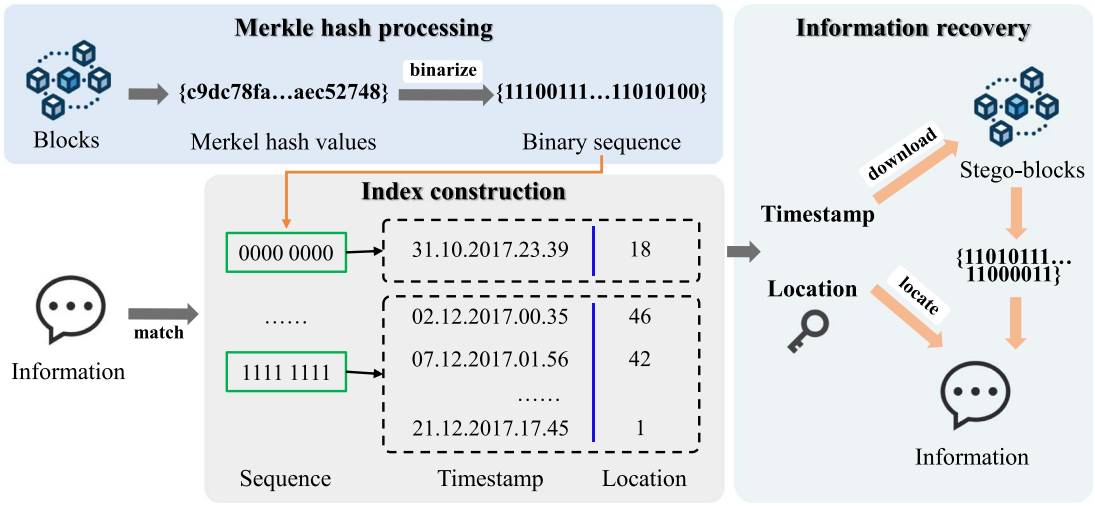

Figure 3. An overview of the proposed BlkInfoM.

Table 1. The notations

Notation	Description
S	Secret info must be matched
$ASCII(S)$	ASCII value of S
BS	Binary S
sg	Segment of BS
n	The number of sg
b	Matched block of sg
bn	The number of picked b
ts	Timestamp of b
mh	Binarized Merkle hash of b
lc	Data location of sg in mh
$B = \{b_1, b_2, ..., b_n\}$	Matched blocks
$TS = \{ts_1, ts_2, ..., ts_m\}$	Timestamps of B
$LC = \{lc_1, lc_2, ..., lc_n\}$	Data locations
$LC' = \{lc_n, lc_{n-1}, ..., lc_1\}$	Encrypted LC

comprehensive elaboration of each step, which also be summarized in Algorithm 1.

1. For secret information S with m characters, we transfer each character into ASCII value, get

$$ASCII(S) = \{as_1, as_2, ...as_m\}, \tag{2}$$

where as is a decimal number.

2. For as_i, convert it to binary numbers with a length of 7-bit, get

$$bs_i = code(as_i). \tag{3}$$

3. Connect bs_i in order, get $BS = \{bs_1, bs_2, ...bs_m\}$, where bs_i is a 7-bit binary sequence, the length of BS is $7 \times m$ bits.
4. BS can eventually be represented as segment $BS' = \{sg_1, sg_2, ...sg_n\}$ according to the following formula:

$$n = \begin{cases} (7 \times m)/8 & \text{if } 7 \times m\%8 = 0, \\ \lceil (7 \times m)/8 \rceil & \text{otherwise.} \end{cases} \tag{4}$$

Algorithm 1 Information-block hiding

Input: Secret information:S
Output: Timestamps: TS ; **Locations**: LC'; **Key**

Transfer S into $ASCII(S) = \{as_1, as_2, ..., as_m\}$
for $as = 1...m$ **do**
  get $bs = code(as)$
end for
Return the list of bs : $BS = \{bs_1, bs_2, ..., bs_m\}$
Divide BS, **get** $\{sg_1, sg_2, ..., sg_n\}$
Get Key
for $sg = 1...n$ **do**
  match sg in index : $match(sg) = (ts, lc)$
end for
Return the lists of ts and lc :
$TS = \{ts_1, ts_2, ..., ts_n\}$, $LC = \{lc_1, lc_2, ..., lc_n\}$
Encrypt LC
Return $LC' = \{lc_n, lc_{n-1}, ..., lc_1\}$
end

After that, each sg represents 8-bit binary sequence, the number of 0 is recorded as key.

5. For sg_i, match it in the constructed index, and get

$$match(sg_i) = (ts_i, lc_i), 1 \le i \le n, \tag{5}$$

$$\begin{aligned} ts_i &= timestamp(b_i), \\ lc_i &= location(mh_i). \end{aligned} \tag{6}$$

Where b_i is the block that sg_i matched, ts_i is the timestamp of b_i; mh_i is the binarized Merkle hash of b_i, lc_i is the data location of sg_i in mh_i.

6. Then, repeat step 5 until all sg are successfully matched, get a set of cryptographic blocks $B = \{b_1, b_2, ...b_n\}$. The binarized Merkle hash of B is denoted as $MH = \{mh_1, mh_2, ...mh_n\}$. Then get the corresponding timestamps and locations:

$$\begin{aligned} TS &= timestamp(B) = \{ts_1, ts_2, ...ts_n\}, \\ LC &= location(MH) = \{lc_1, lc_2, ...lc_n\}. \end{aligned} \tag{7}$$

7. Encrypt LC in reverse order to improve security, get

$$LC^{'} = encrypt(LC) = \{lc_n, lc_{n-1}, ...lc_1\}. \quad (8)$$

8. Transmit TS and $LC^{'}$ with the key.

3.2.2. *Block-information extraction*

After receiving TS and $LC^{'}$ through the public channel, users can extract the information on the basis of the following steps, which also be summarized in Algorithm 2.

Algorithm 2 Block-information extraction

Input: Timestamps : TS ; **Locations** : $LC^{'}$; **Key**

Output: Secret information: S
for $ts = 1...n$ **do**
  download blocks b
end for
Return the lists of $b : B = \{b_1, b_2, ..., b_n\}$
Binarize the mh of $B : MH = \{mh_1, mh_2, ..., mh_n\}$
Decrypt $LC^{'}$, get LC
for $mh = 1...n$ **do**
  extract sg by LC
end for
Connect $sg : BS^{'} = connect(sg_1, sg_2, ..., sg_n)$
Remove the added 0 of $BS^{'}$ by Key, return BS
Divide BS, get $\{bs_1, bs_2, ..., bs_m\}$
for $bs = 1...m$ **do**
  get $as = decode(bs)$
end for
Return the list of $as : ASCII(S) = \{as_1, as_2, ..., as_m\}$
Transfer $ASCII(S)$ into plaintext characters
Return S
end

1. Download the cryptographic blocks (stego-blocks) from blockchain website according to their timestamps $TS = \{ts_1, ts_2, ..., ts_n\}$, get $B = \{b_1, b_2, ..., b_n\}$.
2. For b_i, binarize its Merkle hash values, get a list of binary sequences $MH = \{mh_1, mh_2, ...mh_n\}$.
3. Decrypt $LC^{'}$ in reverse order, get the accurate locations:

$$LC = location(MH) = \{lc_1, lc_2, ...lc_n\}. \quad (9)$$

4. For mh_i, extract the information according to its data location lc_i, get the secret information hided in mh_i:

$$sg_i = extract(mh_i), 1 \le i \le n. \quad (10)$$

5. Then, repeat step 4 until all sg are successfully extracted, connect them to obtain

$$BS^{'} = connect(sg_1, sg_2..., sg_n). \quad (11)$$

6. Remove the added 0 of $BS^{'}$ through key, get the binary secret information BS.
7. Divide BS into 7-bit each, get $BS = (bs_1, bs_2..., bs_m)$.
8. For bs_i, convert it into decimal ASCII value, get

$$as_i = decode(bs_i). \quad (12)$$

The purpose of this step is to convert binary data into recognizable text characters for subsequent processing.

9. Connect as_i in order, get

$$ASCII(S) = \{as_1, as_2, ...as_m\}. \quad (13)$$

This step aggregates individual characters into a complete message sequence.

10. Transfer $ASCII(S)$ into plaintext characters S.

4. BASIC SETUP AND EXPERIMENTAL RESULTS

The experiment was conducted utilizing Python 3.7.5, Intel(R) Core(TM) i7-11700K @ 3.60 GHz, 32.00 GB RAM, and an Nvidia GeForce GTX 3080 Ti GPU. We utilized a dataset of 60 000 blocks, pre-downloaded from a blockchain website, and randomly generated a set of plaintext information to execute matching tests. The process involved hiding and extracting the information within/from these blocks by leveraging timestamps and data locations. Within this section, our analysis primarily focuses on key matching metrics: success rate, false alarm rate, and efficiency, along with an evaluation of the system's security against malicious interceptors and network eavesdroppers.

4.1. Success rate

The success rate of matching is one of the most crucial criteria for evaluating the effectiveness of matching algorithms, directly impacting the feasibility of the entire scheme. In this paper, we adhere to the traditional definition of matching success rate, which can be expressed as:

$$SR = \frac{\sum_1^n f(match_i)}{n} \times 100\%$$
$$f(match_i) = \begin{cases} 1 & \text{if } match(sg_i) = (ts_i, lc_i) \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

In this equation, SR denotes the Success Rate, calculated as the percentage of successful matches. For each match $match_i$, $f(match_i) = 1$ if a match is achieved, and $f(match_i) = 0$ otherwise. The sum of successful matches is divided by the total number n and multiplied by 100% to obtain the success rate. Theoretically, the success rate is influenced by the size of the matching carrier library and the length of the information segments. Therefore, we initiated experiments from these two dimensions and conducted analyses.

4.1.1. *Effect of carrier library size*

Using a pre-downloaded dataset of 60 000 blocks, we selected subsets of 200 to 800 blocks as our libraries. We also created information sequences from 2000 to 10 000 bits for tests, dividing each into 8-bit segments. Experimental results (see Table 2) indicate that, within matching libraries of equal size, the success rate of matching remains relatively stable despite variations in the length of secret information. The success rate effectively increases with the size of the matching library. Based on these

Table 2. Success rate ↑ under different carrier library sizes

Information length	Carrier library size			
	200 blocks (%)	400 blocks (%)	600 blocks (%)	800 blocks (%)
2000-bit	98.8	99.2	99.6	100
4000-bit	97.4	98.8	99.4	99.8
6000-bit	96.3	97.9	98.5	98.9
8000-bit	95.7	97.3	98.4	99.0
10 000-bit	95.1	97.1	98.0	98.6

Table 3. Success rate ↑ under different segment lengths

Information length	Segment length					
	6-bit (%)	8-bit (%)	10-bit (%)	12-bit (%)	14-bit (%)	16-bit (%)
4000-bit	100	99.8	97.8	86.8	60.3	31.2
8000-bit	100	99.0	94.0	80.6	59.9	30.6
16 000-bit	100	99.1	95.1	81.8	59.0	33.3

observations, we hypothesize that a 100% success rate is achievable if secret information can match with a sufficient number of blocks. However, preprocessing a large volume of blocks increases computational costs. We note that for processing 2000 bits of information, preparing 800 blocks suffices not only to achieve a 100% matching success rate but also to avoid excessive memory usage.

4.1.2. *Effect of information segment length*

Using a pre-downloaded dataset of 60 000 blocks, we selected 1000 for our library. We segmented information sequences of 4000, 8000, and 16 000 bits of varying lengths for matching tests. Experimental results (see Table 3) indicate a decline in success rates with increasing segment lengths. This decrease is attributed to the higher probability of matching when secret information is segmented into shorter pieces. Although segmenting information into 6-bit blocks could achieve a 100% success rate, such a strategy proves impractical for real-world applications due to the necessity of downloading and processing a large number of blocks. We observe that segmenting information into 8-bit not only significantly reduces the number of blocks required for processing but also maintains a high success rate.

4.2. False rate

To further validate the robustness of BlkInfoM, we introduce the False Rate (FR) as an additional performance metric. The false alarm rate measures the frequency of invalid matches, where unrelated data is incorrectly identified as a match. This metric is crucial for understanding the system's stability and resilience under varying conditions. The FR calculation formula is as follows:

$$FR = \frac{\sum_1^n g(false_{match_i})}{n} \times 100\%$$
$$g(false_{match_i}) = \begin{cases} 1 & \text{if false match occurs} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

In our framework, FR is defined as 1-SR, reflecting the system's inability to retrieve valid secret segments from pre-registered blockchain blocks. This definition fundamentally differs from conventional false-positive probability in three aspects: (i) No Negative Samples: Our matching process exclusively targets valid secret segments embedded in blockchain transactions (Section 3.1). Non-secret data comparisons are excluded by design. (ii) Deterministic Indexing: The SHA-256 hash index (Section 3.2) creates one-to-one mappings between secret segments and blockchain blocks, preventing false matches to incorrect blocks. (iii) Failure-Only FR: Unsuccessful matches indicate either (a) target block absence in the preloaded library or (b) segment regeneration requirements (Algorithm 2), not random collisions. This operational definition aligns with blockchain's inherent anti-collision properties. As Bitcoin consensus rules prohibit duplicate transaction metadata [22], false-positive matches become theoretically impossible in properly configured systems.

We analyze the false alarm rate in two dimensions: the size of the carrier library and the length of the information segment. Specifically, we selected a 4000-bit secret message, divided it into segments of varying lengths for testing, and used a pre-downloaded dataset of 60 000 blocks, selecting subsets of 200 to 800 blocks as our library. The experimental results, shown in Table 4, indicate that within libraries of the same size, the FR increases with the length of the segment. This increase is attributed to the higher probability of false matches as secret information is divided into longer segments. In contrast, for segments of the same length, the FR decreases as the size of the matching library increases.

These findings suggest that an optimal balance between segment length and library size is essential to maintain a low false alarm rate. Although shorter segments reduce the risk of false matches, they require a larger carrier library to maintain high success rates. However, increasing the size of the library allows for greater tolerance with longer segments, which increases the system flexibility. This balance is critical for BlkInfoM's adaptability in various real-world scenarios, as it enables system configuration adjustments based on available resources and desired performance levels.

The experimental analysis also reveals one counterintuitive but valid phenomenon: While 6-bit segments theoretically have higher collision probability, our implementation achieves 0% FR due to (i) Structural constraints in Bitcoin headers limiting actual 6-bit combinations to 38 unique patterns (vs. 64

Table 4. False alarm rate ↓ under different carrier library sizes and segment lengths

Carrier library size	Segment length					
	6-bit (%)	8-bit (%)	10-bit (%)	12-bit (%)	14-bit (%)	16-bit (%)
200 blocks	0.6	2.6	3.2	15.3	40	69.1
400 blocks	0.2	1.2	2.7	7.1	32.9	47.6
600 blocks	0	0.6	1.6	15.7	28.9	37.2

Table 5. Time cost (minutes) ↓ under different information lengths and library sizes

Information length	Index	Carrier library size			
		200 blocks	400 blocks	600 blocks	800 blocks
2000-bit	✓	0.13	0.13	0.13	0.15
	×	0.26	0.41	0.72	0.93
6000-bit	✓	0.35	0.4	0.45	0.45
	×	0.76	1.02	1.51	2.01
10 000-bit	✓	0.5	0.5	0.5	0.5
	×	1.04	1.53	2.02	2.93

theoretical) in 60 000 blocks; (ii) Hash-indexed direct addressing eliminating probabilistic scanning. These findings demonstrate how blockchain's unique properties reshape traditional steganographic performance tradeoffs. The FR metric in BlkInfoM primarily reflects system configuration completeness rather than algorithmic limitations.

4.3. Efficiency

Efficiency is consequential in this era with vast amounts of data as people don't want to spend a lot of time retrieving information [53]. In BlkInfoM, the primary time cost is attributed to the querying process for mapping binary information to block data. Theoretically posited, the index structure we constructed can significantly reduce this overhead. To validate the efficiency of BlkInfoM and underscore the necessity of the index structure, we conducted matching tests on 2 000 to 10 000 bits of information, both with/without the index, across block libraries of 200, 400, 600, and 800 blocks. Each experiment was conducted five times, with the average outcomes recorded in Table 5 for comprehensive analysis.

Experimental results reveal BlkInfoM's small time overhead under standard settings, and while the cost slightly increases with the length of the information, this increment is within expectations. Notably, the absence of an index structure makes the size of the block library significantly affect efficiency—searching a large database for specific information is like searching for a needle in a haystack, with smaller libraries enabling faster searches. Introducing an index structure enhances efficiency across all library sizes by enabling direct access to relevant blocks. Our experiments highlight the index's key role in boosting efficiency by correlating information length with library size.

4.4. Security analysis against interceptors and eavesdroppers

In response to potential attacks by malicious interceptors and network eavesdroppers, we further expand the security analysis of BlkInfoM. While a direct quantitative or qualitative analysis of BlkInfoM's resistance to these threats is challenging, we aim to provide as comprehensive a discussion as possible regarding its effectiveness in mitigating these security risks.

BlkInfoM inherently strengthens resistance to interception and eavesdropping by relying on immutable blockchain data and key-based retrieval mechanisms. Even in scenarios where timestamps and location data might be intercepted, the absence of complete mapping rules significantly limits an unauthorized entity's ability to identify mapped information on the blockchain. Additionally, the anonymity and irreversibility of blockchain transactions further impede successful decoding of secret information in the event of network eavesdropping.

5. BLKINFOM APPLICATIONS

Beyond the validated high matching success rate and efficiency, to further examine the practicality of BlkInfoM, we applied it within coverless steganography for information transmission. Besides, considering storage needs and cost benefits, we also merged it with zero-watermarking technology to build a watermark storage platform for copyright protection. For coverless information transmission, we have provided a thorough *security analysis* covering anti-steganalysis, anti-attack, and anti-reveal properties, along with a comprehensive *hiding capacity* comparison against traditional methods. For zero-watermark copyright protection, we highlighted BlkInfoM's *cost-efficiency* and *reliability* as a local watermark storage and retrieval platform.

5.1. Coverless information transmission

Similar to the mapping steps explored in Section 3.2, we deploy the BlkInfoM mechanism for the task of coverless information transmission. Figure 4 illustrates the result of hiding and extracting secret information. The foremost benefit of employing BlkInfoM for coverless information transmission is its enhanced security and significant capacity.

5.1.1. *Security analysis*

We rigorously demonstrate the security of our proposed solution from three perspectives: i Anti-steganalysis. By leveraging a

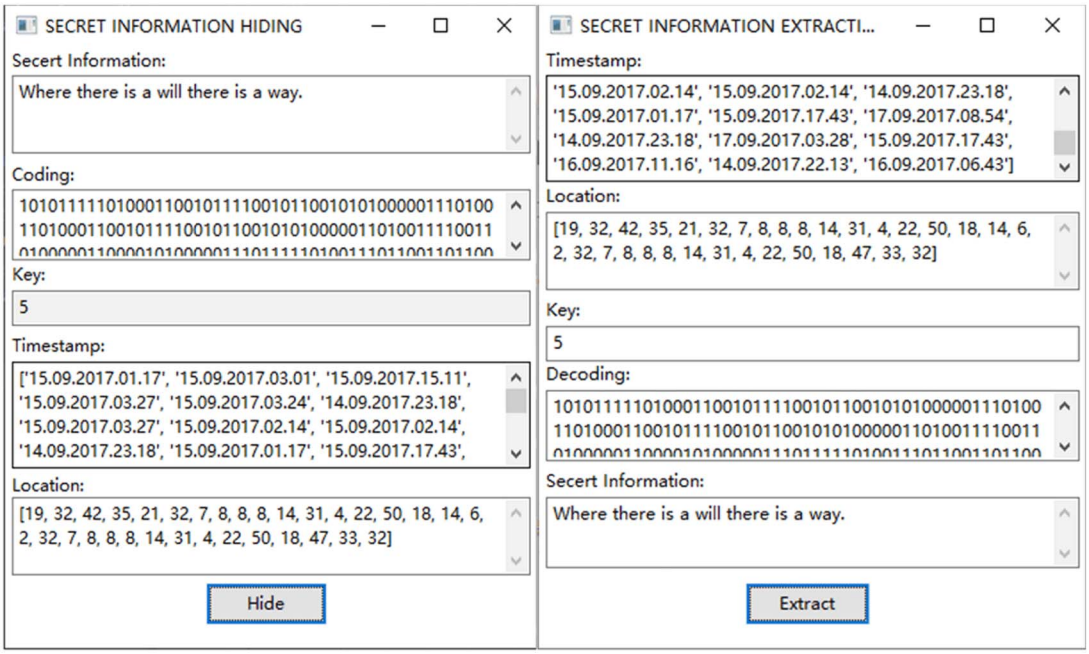

Figure 4. Utilizing BlkInfoM, plaintext secret information can be accurately transmitted through a key containing timestamps and location information.

Table 6. Maximum capacity comparison

Method	Image-based					Ours
	Pixel	SIFT+BOF	DCT	DWT	CNN	
Capacity ↑	8-bit	8-bit	15-bit	15-bit	6 × N-bit	63-bit

coverless mapping mechanism based on existing blockchain transactions, our approach circumvents steganalysis tools without altering carriers, enhancing concealment and minimizing data redundancy. ii Anti-attack. Utilizing the immutable nature of blockchain transaction data, our method ensures stable information transmission. Once a mapping is established, secret information can be securely retrieved using a key, without the risk of alterations or loss due to attacks. iii Anti-reveal. Our approach makes it challenging for unauthorized entities to pinpoint the mapped information on the blockchain without the key. Even if timestamps and location data are compromised, the lack of detailed mapping rules and the anonymity of blockchain transactions prevent access to the secret information.

5.1.2. *Capacity comparison*

Capacity stands as a critical metric due to the limited information transmission capability of current mapping carriers. Typically, in coverless image steganography, capacity is defined as the maximum information length that an image can map. Our study contrasts our approach with several coverless image steganography techniques. As shown in Table 6, our method allows a processed block to represent up to 63-bit binary sequences, while Pixel-based [9], and SIFT+BOF-based [54] approach divides each image into 3 × 3 blocks, generating an 8-bit binary sequence per block, DCT-based [10] and DWT-based [15] method generates binary sequences ranging from 1 to 15 bits using transformation algorithms, CNN-based method [14] hides information by selecting objects detected in an image (numbering N), where each object represents 6-bit. Evidently, compared to images, the block transaction data we utilize offers a significant advantage in mapping capacity.

Despite a block's capability to represent up to 63-bit sequences, not all sequences may map successfully to secret information, and some sequences might be matched multiple times. To objectively assess the capacity of our method, we adopted a capacity measurement approach similar to text-based coverless steganography, focusing on the number of secret information segments a block can match, as follows:

$$C = n/bn, bn \leq n \tag{16}$$

where n represents the number of secret information segments, the bn is the number of blocks used for matching, like the concept of keywords and texts used in text-based matching. As illustrated in Fig. 5, our method's capacity increases with the length of the secret information. When hiding secret information of 8000 bit, each block can match more than 4 secret information segments. For fair capacity comparison, we use the mean capacity as a metric, which is the average of 100 capacities when n is in the range from 8 to 800. The comparison results with two other coverless text steganography techniques [35, 36] that use keyword mapping are shown in Table 7, where our method demonstrates a higher capacity.

To provide a more comprehensive evaluation, we further compared the capacity of BlkInfoM with three blockchain-based covert communication methods (aZhang *et al.* [51], bZhang *et al.* [55], and Cao *et al.* [56]) in terms of average embedding capacity per transaction, as shown in Fig. 6. Compared to other blockchain-based methods, our approach achieves higher embedding efficiency by

Table 7. Mean capacity comparison

Method	Text-based		Ours
	Chinese	Multi-keywords	
Capacity ↑	1	1.57	2.83

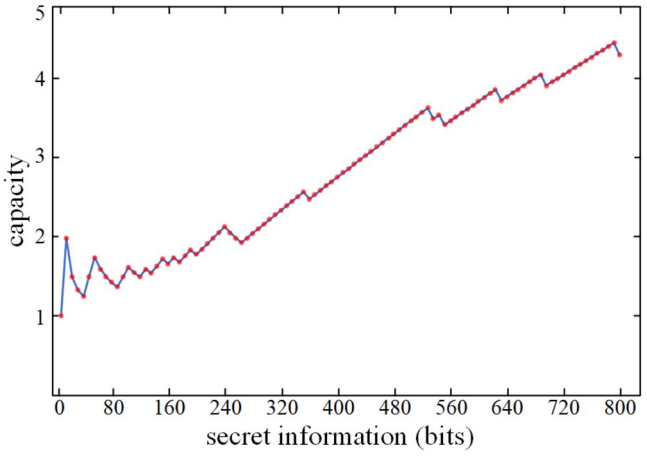

Figure 5. The capacity of BlkInfoM-based coverless steganography.

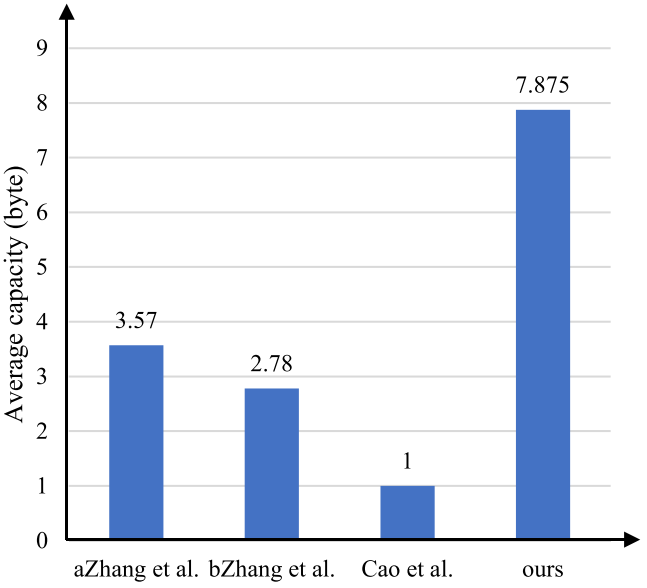

Figure 6. Average embedding capacity comparison with Blockchain-based methods.

directly utilizing existing transaction data on the blockchain, rather than relying on the creation of new transactions. This independence from transaction generation makes BlkInfoM both more flexible and efficient.

These results collectively highlight BlkInfoM's superiority in capacity over both traditional and blockchain-based methods, confirming its scalability and efficiency in real-world covert communication and data protection scenarios.

5.1.3. *Case study*

We simulate three covert communication scenarios to showcase BlkInfoM's advantages in secure, high-capacity, and real-time environments.

Case Study 1: Secure Communication over Public Networks. In public network environments, conventional steganographic methods are vulnerable to detection due to carrier modifications. BlkInfoM addresses this by mapping encrypted secrets onto immutable blockchain data, eliminating the need for visible or editable carriers. Because the carrier data is pre-existing and unaltered, the risk of detection is significantly reduced. Even if the transmission is intercepted, an adversary without the mapping key cannot extract meaningful content. Experimental results show that the retrieval accuracy exceeds 99.8% with negligible failure rate (see Section 4.1), demonstrating the system's robustness for high-security applications.

Case Study 2: High-Capacity Covert Storage. Covert operations often demand the ability to store large volumes of secret information. We tested BlkInfoM's capacity by encoding sequences up to 10 000 bits, using blockchain transaction hashes as carriers. Each block successfully supported up to 63-bit encodings. Compared to traditional coverless steganography methods (e.g. CNN- or DCT-based approaches), BlkInfoM achieved a 4–8× improvement in per-block payload size (see Table 7). Furthermore, its indexed hash retrieval structure enables efficient decoding without the need to scan the entire blockchain, making it scalable for large covert storage tasks.

Case Study 3: Real-Time Dynamic Communication. In real-time environments such as sensor networks, battlefield communications, or streaming platforms, the timely and covert transmission of data is critical. BlkInfoM supports sub-second retrieval latency by leveraging pre-indexed blockchain structures, as confirmed by our time-cost evaluation (Table 6). Even under dynamically changing inputs, the retrieval mechanism remained efficient regardless of library size. This proves the system's capability to support low-latency, real-time covert messaging.

5.2. Zero-watermark copyright protection

Beyond covert communication, BlkInfoM is also designed to serve as a decentralized, cost-effective platform for robust copyright protection using zero-watermarking techniques. As illustrated in Fig. 7 the process begins with ③ retrieving the encrypted copyright identifier via ④ a secure key. This key is then combined with ② the publicly available image information, and through reverse reconstruction using zero-watermarking principles, ① the author's identity is accurately recovered. This process enables precise and tamper-proof copyright verification.

Thanks to its high matching success rate (see Section 4.1), BlkInfoM ensures reliable key-based retrieval of copyright information, which is critical for preventing legal disputes and unauthorized usage. Furthermore, BlkInfoM is deployed locally, thereby eliminating the need for third-party storage services. This self-contained approach significantly reduces operational costs and minimizes the risk of data leakage, providing a secure and efficient alternative to traditional copyright protection systems.

To further validate the system's practical utility in this domain, we present the following three case studies that demonstrate BlkInfoM's capabilities across different copyright-related scenarios.

Case Study 4: Digital Artwork Copyright Verification. To verify authorship of digital images, we applied BlkInfoM to generate a unique zero-watermark key without modifying the image itself. This key was mapped to blockchain transaction data (timestamps and locations), and the original image was later used with the key to reconstruct the author's information. In testing, the retrieval success rate reached 100% under normal conditions. This enables

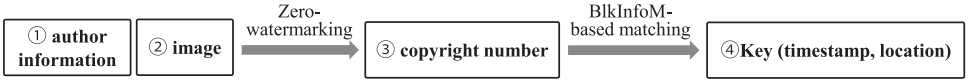

Figure 7. Illustration of using BlkInfoM for watermark storage and retrieval to protect copyrights.

secure, tamper-free copyright verification, which is especially valuable in legal dispute scenarios.

Case Study 5: Scalable Copyright Management for Large Image Sets. We evaluated BlkInfoM's scalability on a dataset containing over 5000 images. By storing copyright keys locally and avoiding third-party platforms, BlkInfoM ensured full data sovereignty. The indexed mapping architecture allowed metadata for all images to be efficiently stored and retrieved without performance degradation. This proves the feasibility of deploying BlkInfoM for enterprise-scale copyright management at minimal operational cost.

Case Study 6: Real-Time Copyright Authentication. In content-driven platforms like social media, timely copyright verification is crucial before public release. We simulated an authentication pipeline where digital content was checked in real time. BlkInfoM's precomputed index table allowed copyright verification to be completed in under 0.5 seconds per query. This demonstrates the system's capability to provide immediate copyright enforcement, meeting the demands of high-speed, content-intensive environments.

5.3. Challenges and solutions in practical deployment

While BlkInfoM has demonstrated promising performance in controlled experimental settings, several technical challenges may arise during real-world deployment. These include handling the dynamic nature of blockchain data, managing large-scale storage demands, and ensuring efficient computation under time constraints. We have analyzed these issues in depth and propose the following corresponding strategies:

5.3.1. *Real-time data handling*

The dynamic and constantly growing nature of blockchain transactions may affect query latency and data-matching performance. To mitigate this, we propose the construction of an *index-based retrieval structure* that enables fast lookup of relevant Merkle root segments and associated metadata. In addition, *distributed caching* mechanisms can be introduced to store frequently accessed block segments in memory, significantly reducing disk I/O and improving query response times under high-concurrency conditions.

5.3.2. *Storage capacity management*

Given the high volume and redundancy of blockchain data, efficient storage management is crucial. We recommend the application of *lightweight compression schemes* (e.g. delta encoding or header-only extraction) to retain only essential fields (such as block headers and Merkle root values), thereby reducing the storage footprint. Furthermore, by adopting a *dynamic block selection policy* based on usage frequency and data relevance, the system can avoid unnecessary full-chain storage while preserving functionality and completeness.

5.3.3. *Processing speed optimization*

Large-scale data matching tasks may lead to computational bottlenecks if processed sequentially. To address this, we suggest employing *parallel processing frameworks* such as MapReduce or multi-threaded hash indexing, which allow concurrent matching operations. Moreover, BlkInfoM's modular architecture can be extended to support *distributed task scheduling*, enabling scalable deployment across multiple nodes and further improving throughput and fault tolerance.

Through these targeted strategies, BlkInfoM can be effectively adapted to diverse deployment environments while maintaining performance, reliability, and scalability. These measures not only enhance system robustness in real-world conditions but also offer actionable guidelines for future implementations.

6. CONCLUSIONS

In this work, we exploit the block structure and rich&reliable transaction data of Blockchain to develop an innovative mapping mechanism, BlkInfoM, which supports both coverless information transmission and zero-watermark copyright protection through accurate information-to-block matching. Experiments and analysis confirm BlkInfoM's exceptional performance and practicality, offering an enlightening strategy for integrating blockchain technology into information hiding practices.

ACKNOWLEDGEMENTS

This work was supported in part by the Postgraduate Research and Innovation Project of Hunan Province under Grant CX20220049.

DATA AVAILABILITY

The data underlying this article were derived from sources in the public domain [https://www.blockchain.com/explorer]. The authors will supply the relevant data in response to reasonable requests.

REFERENCES

1. Shen W, Qin J, Yu J. *et al.* Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. *IEEE Trans Inf Forensics Secur* 2018;**2**:331–46.
2. Delina B. Information hiding: a new approach in text steganography. In: JMZS de la Maza, Espi PLL (eds.), *Proceedings of WSEAS 2008. Stevens Point, WI, 24–26 June*. New York: ACM, 2008, 689–95.
3. Mielikainen J. LSB matching revisited. *IEEE Signal Process Lett* 2006;**13**:285–7.
4. Jafar IF, Darabkh KA, Al-Zubi RT. *et al.* Efficient reversible data hiding using multiple predictors. *Comput J* 2016;**59**:423–38.
5. Liu Y, Liu S, Wang Y. *et al.* Video steganography: a review. *Neurocomputing* 2019;**335**:238–50.
6. Qin J, Luo Y, Xiang X. *et al.* Coverless image steganography: a survey. *IEEE access* 2019;**7**:171372–94.
7. Chen C, Shi YQ. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Proceedings of ISCAS 2008, Seattle, WA, 18–21 May*. Piscataway: IEEE, 2008, 3029–32.
8. Yang Z, Yang H, Chang CC. *et al.* Real-time steganalysis for streaming media based on multi-channel convolutional sliding windows. *Knowl Based Syst* 2022;**237**:107561.

9. Zhou Z, Sun H, Harit R. *et al.* Coverless image steganography without embedding. In: Zhiqiu H, Xingming S, Junzhou L, Jian W (eds.), *Proceedings of ICCCS 2015, Nanjing, China, 13–15 August*. Berlin: Springer, 2015, 123–32.
10. Zhang X, Peng F, Long M. Robust coverless image steganography based on DCT and LDA topic classification. *IEEE Trans Multimedia* 2018;**20**:3223–38.
11. Chang CC, Lin PY. Adaptive watermark mechanism for rightful ownership protection. *J Syst Softw* 2008;**81**:1118–29.
12. Luo Y, Qin J, Xiang X. *et al.* Coverless real-time image information hiding based on image block matching and dense convolutional network. *J Real-time Image Pr* 2020;**17**:125–35.
13. Luo Y, Qin J, Xiang X. *et al.* Coverless image steganography based on image segmentation. *CMC-Comput Mater Con* 2020;**64**:1281–95.
14. Luo Y, Qin J, Xiang X. *et al.* Coverless image steganography based on multi-object recognition. *IEEE Trans Circuits Syst Video Technol* 2020;**31**:2779–91.
15. Liu Q, Xiang X, Qin J. *et al.* Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping. *Knowl Based Syst* 2020;**192**:105375.
16. Liu Q, Xiang X, Qin J. *et al.* A robust coverless steganography scheme using camouflage image. *IEEE Trans Circuits Syst Video Technol* 2021;**32**:4038–51.
17. Chen TH, Horng G, Lee WB. A publicly verifiable copyright-proving scheme resistant to malicious attacks. *IEEE Trans Ind Electron* 2005;**52**:327–34.
18. Dong P, Brankov JG, Galatsanos NP. *et al.* Digital watermarking robust to geometric distortions. *IEEE Trans Image Process* 2005;**14**:2140–50.
19. Lai CC, Tsai CC. Digital image watermarking using discrete wavelet transform and singular value decomposition. *IEEE T Instrum Meas* 2010;**59**:3060–3.
20. Yang K, Wang W, Yuan Z. *et al.* Strong robust zero watermarking algorithm based on NSCT transform and image normalization. In: Jun W, Xingming S, Jian W, Jianhua Z (eds.), *Proceedings of IAEAC 2018, Chengdu, China, 20–22 March*. Piscataway: IEEE, 2018, 236–240.
21. Wang R, Shaocheng H, Zhang P. *et al.* A novel zero-watermarking scheme based on variable parameter chaotic mapping in NSPD-DCT domain. *IEEE Access* 2020;**8**:182391–411.
22. Yang C, Li J, Bhatti UA. *et al.* [retracted] robust zero watermarking algorithm for medical images based on Zernike-DCT. *Secur Commun Netw* 2021;**2021**:4944797.
23. Ren N, Guo S, Zhu C. *et al.* A zero-watermarking scheme based on spatial topological relations for vector dataset. *Expert Syst Appl* 2023;**226**:120217.
24. Crosby M, Pattanayak P, Verma S. *et al.* Blockchain technology: beyond bitcoin. Applied innovation. *Applied Innovation Review* 2016;**2**:71.
25. Wang S, Ouyang L, Yuan Y. *et al.* Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE T Syst Man Cy-s* 2019;**49**:2266–77.
26. Hackius N, Petersen M. Blockchain in logistics and supply chain: Trick or treat?. In digitalization in supply chain management and logistics: Smart and digital solutions for an industry 4.0 environment. In: Carlos J, Wolfgang K, Christian MR (eds.), *Proceedings of HICL 2017, Hamburg, Germany, 12–13 October*. Berlin: epubli GmbH, 2017, 3–18.
27. Li R, Song T, Mei B. *et al.* Blockchain for large-scale internet of things data storage and protection. *IEEE Trans Serv Comput* 2018;**12**:762–71.
28. Zyskind G, Nathan O. Decentralizing privacy: Using blockchain to protect personal data. In: Sven D, Anna S, Sean P, Matt B (eds.), *Proceedings of SPW 2015, San Jose, CA, 21 May*. Piscataway: IEEE, 2015, 180–4.
29. Goodfellow I, Pouget-Abadie J, Mirza M. *et al.* Generative adversarial nets. *Adv Neural Inf Process Syst* 2014;**27**2672–80.
30. Hu D, Wang L, Jiang W. *et al.* A novel image steganography method via deep convolutional generative adversarial networks. *IEEE access* 2018;**6**:38303–14.
31. Chen X, Zhang Z, Qiu A. *et al.* Novel coverless steganography method based on image selection and StarGAN. *IEEE Trans Netw Sci Eng* 2020;**9**:219–30.
32. Li Q, Wang X, Wang X. *et al.* An encrypted coverless information hiding method based on generative models. *Inform Sci* 2021;**553**:19–30.
33. Peng F, Chen G, Long M. A robust coverless steganography based on generative adversarial networks and gradient descent approximation. *IEEE Trans Circuits Syst Video Technol* 2022;**32**:5817–29.
34. Wang K, Gao Q. A coverless plain text steganography based on character features. *IEEE Access* 2019;**7**:95665–76.
35. Zhou Z, Mu Y, Yang CN. *et al.* Coverless multi-keywords information hiding method based on text. *Int J Secur Appl* 2016;**10**:309–20.
36. Chen X, Sun H, Tobe Y. *et al.* Coverless information hiding method based on the Chinese mathematical expression. In: Zhiqiu H, Xingming S, Junzhou L, Jian W (eds.), *Proceedings of ICCCS 2015, Nanjing, China, 13–15 August*. Berlin: Springer, 2015, 133–43.
37. Tan Y, Qin J, Xiang X. *et al.* Coverless steganography based on motion analysis of video. *Secur Commun Netw* 2021;**2021**:5554058.
38. Luo Y, Zhou T, Liu F. *et al.* IRWArt: Levering watermarking performance for protecting high-quality artwork images. In: Ying D, Jie T (eds.), *Proceedings of WWW 2023, Austin, Texas, 30 April-4 May*. New York: ACM, 2023, 2340–8.
39. Fahmy H, El-Gendy EM, Mohamed MA. *et al.* ECH3OA: an enhanced chimp-Harris hawks optimization algorithm for copyright protection in color images using watermarking techniques. *Knowl Based Syst* 2023;**269**:110494.
40. Wen Q, Sun T, Wang S. Concept and application of zero-watermark. *Acta Electonica Sinica* 2003;**31**:214.
41. Ren N, Zhao Y, Zhu C. *et al.* Copyright protection based on zero watermarking and blockchain for vector maps. *ISPRS Int J Geoinf* 2021;**10**:294.
42. Wu X, Ma P, Jin Z. *et al.* A novel zero-watermarking scheme based on NSCT-SVD and blockchain for video copyright. *EURASIP J Wirel Comm* 2022;**2022**:20.
43. Xu D, Zhu C, Ren N. A zero-watermark algorithm for copyright protection of remote sensing image based on blockchain. In *Proceedings of ICBCTIS 2022, Nanjing, China, 14–16 October*. Piscataway: IEEE, 2022, 111–6.
44. Wang B, Jiawei S, Wang W. *et al.* Image copyright protection based on blockchain and zero-watermark. *IEEE Trans Netw Sci Eng* 2022;**9**:2188–99.
45. Truong NB, Sun K, Lee GM. *et al.* Gdpr-compliant personal data management: a blockchain-based solution. *IEEE Trans Inf Forensics Secur* 2019;**15**:1746–61.
46. She W, Huo L, Tian Z. *et al.* A double steganography model combining blockchain and interplanetary file system. *Peer Peer Netw Appl* 2021;**14**:3029–42.
47. Meng Z, Morizumi T, Miyata S. *et al.* Design scheme of copyright management system based on digital watermarking and blockchain. In: Hua M, Kamrul H (eds.), *Proceedings of COMPSAC 2018, Tokyo, Japan, 23–27 July*. Piscataway: IEEE, 2018, 359–64.

48. Tschorsch F, Scheuermann B. Bitcoin and beyond: a technical survey on decentralized digital currencies. *IEEE Commun Surv Tut* 2016;**18**:2084–123.
49. Vasek M, Moore T. There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In: Rainer B Tatsuaki O (eds.), *Proceedings of FC 2015, San Juan, Puerto Rico, 26–30 January*. Berlin: Springer, 2015, 44–61.
50. Zhang L, Zhang Z, Wang W. *et al.* Research on a covert communication model realized by using smart contracts in blockchain environment. *IEEE Syst J* 2021;**16**:2822–33.
51. Zhang P, Cheng Q, Zhang M. *et al.* A blockchain-based secure covert communication method via shamir threshold and stc mapping. *IEEE Trans Dependable Secure Comput* 2024;**21**:4469–80.
52. Heilman E, Alshenibr L, Baldimtsi F. *et al.* Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. In: Hilarie O (ed.), *Proceedings of NDSS 2017, San Diego, CA, 2 February-1 March*. Reston: ISOC, 2017.
53. Sharma VK, Mir RN. An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm. *J King Saud Univ-Com* 2022;**34**:615–26.
54. Yuan C, Xia Z, Sun X. Coverless image steganography based on SIFT and BOF. *J Internet Technol* 2017;**18**:435–42.
55. Zhang P, Cheng Q, Zhang M. *et al.* A group covert communication method of digital currency based on blockchain technology. *IEEE Trans Netw Sci Eng* 2022;**9**:4266–76.
56. Cao H, Yin H, Gao F. *et al.* Chain-based covert data embedding schemes in blockchain. *IEEE Internet Things J* 2020;**9**:14699–707.

The Computer Journal, 2025, **68**, 1967–1979
https://doi.org/10.1093/comjnl/bxaf088
Original Article