Research on Covert Communication Technology Based on Matrix Decomposition of Digital Currency Transaction Amount

Lejun Zhang[1, 2, 3, 4*], Bo Zhang[1], Ran Guo[5*], Zhujun Wang[1], Guopeng Wang[3*], Jing Qiu[2], Shen Su[2], Yuan Liu[2], Guangxia Xu[2], Zhihong Tian[2], and Sergey Gataullin[6, 7]

[1] College of Information Engineering, Yangzhou University, Yangzhou 225127, China
[2] The Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
[3] Research and Development Center for E-Learning, Ministry of Education, Beijing, 100039, China
[4] School Math & Computer Science, Quanzhou Normal University, Quanzhou 362000, China
[5] School of Physics and Materials Science, Guangzhou University, Guangzhou 510006, China
[6] Central Economic and Mathematics Institute, Russian Academy of Sciences, Moscow, Russia
[7] MIREA - Russian Technological University, 78 Vernadsky Avenue, Moscow, Russian Federation119454
[e-mail: zhanglejun@yzu.edu.cn, MZ120210953@yzu.edu.cn, guoran@gzhu.edu.cn, DX120210091@yzu.edu.cn, wangguopeng@sohu.com, qiujing@gzhu.edu.cn, sushen@gzhu.edu.cn, liuyuan@swc.neu.edu.cn, xugx@cqupt.edu.cn, tianzhihong@gzhu.edu.cn, sgataullin@cemi-ras.ru]
[*]Corresponding author: Lejun Zhang, Ran Guo, Guopeng Wang

Received September 17, 2023; revised January 5, 2024; accepted March 11, 2024; published April 30, 2024

Abstract

With the development of covert communication technologies, the number of covert communication technologies using blockchain as a carrier is increasing. However, using the transaction amount of digital currency as a carrier for covert communication has problems such as low embedding rate, large consumption of transaction amount, and easy detection. In this paper, firstly, by experimentally analyzing the distribution of bitcoin transaction amounts, we determine the most suitable range of amounts for matrix decomposition. Secondly, we design a novel matrix decomposition method that can successfully decompose a large amount matrix into two small amount matrices and utilize the elements in the small amount matrices for covert communication. Finally, we analyze the feasibility of the novel matrix decomposition method in this scheme in detail from four aspects, and verify it by experimental comparison, which proves that our scheme not only improves the embedding rate and reduces the consumption of transaction amount, but also has a certain degree of resistance to detection.

Keywords: blockchain, covert communication, matrix decomposition, transaction amount, embedding rate, detection resistance

This work is sponsored by the National Natural Science Foundation of China No. 62172353, No. 62302114, No. U20B2046 and No. 61976064. Future Network Scientific Research Fund Project No. FNSRFP-2021-YB-48. Innovation Fund Program of the Engineering Research Center for Integration and Application of Digital Learning Technology of Ministry of Education No.1221045.

http://doi.org/10.3837/tiis.2024.04.011 **ISSN : 1976-7277**

1. Introduction

Covert communication [1] is a method of hiding information, a process in which the sender transmits a covert message over a public channel through a carrier and the receiver receives the message, making it difficult for a third party to notice or detect the existence of the communication. In traditional covert communication, images [2], videos [3], files [4] , etc. are often used as carriers. Although they have a high embedding rate, the content of the communication is susceptible to problems such as eavesdropping. Blockchain technology is characterized by decentralization and anonymity. The choice of blockchain as a carrier for covert communications helps to compensate for some of the deficiencies that exist in traditional covert communications, but it is also necessary to consider the security and privacy challenges specific to blockchain.

Blockchain covert communication refers to the use of covert communication technology for secure and hidden messaging within the framework of blockchain technology. In this scenario, both communicators utilize a certain carrier of the blockchain to communicate while keeping their communication activities hidden from outside observers. Although this provides an additional layer of security for communication, however, the main challenge in blockchain covert communication is the balance between covertness and embedding rate. Concealment requires that communication activities are not readily detectable, while the embedding rate relates to the ability to embed information in the communication vector. In blockchain covert communication, potential vectors include transaction amounts [5], storage fields [6], transaction addresses [7], protocols [8], etc. in blockchain transactions. However, blockchain transaction amount as one of the carriers has some problems, such as low embedding rate, large transaction amount, and easy detection. The number of embedded messages increases as the transaction amount becomes larger, but a high transaction amount also increases the risk of being easily detected.

Therefore, this paper designs a scheme to realize covert communication based on digital currency transaction amount matrix decomposition method. First, the information to be transmitted is encrypted to form a ciphertext, and then the ciphertext is sliced and transformed into a large amount matrix. Next, two small transaction amount matrices are obtained by decomposing the large amount matrix. Finally, we skillfully utilize the amounts in these two small matrices as transmission carriers to transmit the information. This scheme not only improves the embedding rate and reduces the consumption of transaction amounts, but also has certain resistance to detection.

Our contribution can be summarized as follows.

• In this paper, we design a novel matrix decomposition method, which actually represents the large amount matrix by two small amount matrices. The method reduces the amount consumption and improves the embedding rate. And the method makes the distribution of the length of the amount in the decomposition matrix on the range d in this method, which makes the scheme resistant to detection.

• The set of transaction sender addresses and the set of transaction receiver addresses are generated by sharing Key off-chain. And the positions of the elements in the small amount matrix are represented by the interaction between the addresses.

• Finally, this paper demonstrates the feasibility of the novel matrix decomposition method and analyzes the resistance to detection and embedding rate of this scheme by experiments.

The rest of the paper is organized as follows. In Section 2, the research work related to blockchain covert communication, common matrix decomposition methods and the matrix decomposition method designed in this paper are introduced. Section 3 details the scheme

design for implementing covert communication based on the matrix decomposition method of digital currency transaction amounts. In Section 4, we first compare and analyze the matrix decomposition method from four aspects, then we analyze the resistance to detection and embedding rate of this scheme, and finally we compare and analyze the performance with other schemes. The last section concludes the whole paper.

2. Related Work

In recent years, with the development of metaverse [9], research on blockchain carriers [10][11] for covert communication as its underlying technology is increasing.

Partala J.[12] first proposed the blockchain covert communication scheme (BLOCCE), which transmits messages embedded in the least significant bit (LSB) of an address. Zhang L et al.[13] proposes the method of covert communication (V_BLOCCE), which is a scheme that realizes covert communication by embedding covert information into addresses through the Vanitygen generate special address tool. Zhang L et al.[14] proposed a scheme to utilize fields in the whisper [15] protocol for covert message transfer in an Ethereum environment. Tian J et al.[16] proposed a blockchain covert communication scheme based on dynamic labels (DLchain), which is to store the dynamic labels in the OP_RETURN field, and the covert messages are embedded in the signatures of multiple transactions respectively for covert transmission. Gao F et al.[17] proposed a scheme which is based on Kleptography [18] algorithm to embed the covert information into signatures for covert information transmission.

There are problems such as changing the structure of the transaction in the above schemes. In the current scheme, the transaction structure is easily changed, so the researchers propose a covert communication scheme with the transaction amount of the digital currency. Liu S et al. [19] proposed three embedding approaches for building blockchain covert communication in Ethereum using the Value field of a transaction [20], which are One-Bit Embedding scheme (OBE), HMAC-based Multi-Bit Embedding scheme (HMAC-based MBE), and Hash-based Multi-Bit Embedding scheme (Hash-based MBE). Luo X et al.[21] propose a novel covert communication scheme based on Bitcoin transactions (NCCM), which involves embedding messages onto the interactions between Bitcoin transaction amounts and addresses for covert transmission. Akbari I et al.[22] proposed a combination of transaction and image steganography (TISCC), where off-chain steganography is done via images and on-chain smart contracts are invoked to transmit the steganographic information in the transaction field in Ethereum. However, the problems of large transaction amounts, low embedding rates and easy detection in schemes that use the transaction amounts of digital currencies as carriers for covert communication have not been properly addressed.

Matrix decomposition [23] is to split a matrix into the product of multiple matrices. Therefore, it is able to decompose a large-amount matrix into two small-amount matrices, effectively solving the problems of too large transaction amounts and low embedding rates. The methods of matrix decomposition include Triangular Decomposition [24], Full Rank Decomposition [25], QR Decomposition [26], and SVD (Singular Value) Decomposition [27]. However, the matrices obtained from these decompositions are usually uncontrollable and may have non-positive integers, random lengths, etc., which do not meet the requirements of this paper's scheme for setting the transaction amount. Therefore, in this paper, a novel matrix decomposition method is designed, which is able to decompose a large amount (integer) matrix into the product of two small amount (integer) matrices, i.e., $Matrix_{Number} = Matrix_a \times Matrix_b$, as shown in (1) and (2). In this equation, the diagonal of $Matrix_{Number}$, and the elements in the $Matrix_a$ and $Matrix_b$ matrices represent the transaction amount. The matrix

decomposition method designed in this scheme can effectively control the length of the elements in the decomposed small amount matrix, thus solving the problem of easy detection.

$$Matrix_{Number} = Matrix_a \times Matrix_b \tag{1}$$

$$\begin{pmatrix} Number_0 & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & Number_n \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ \vdots & \vdots \\ a_{n0} & a_{n1} \end{pmatrix} \times \begin{pmatrix} b_{00} & \cdots & b_{0n} \\ b_{10} & \cdots & b_{1n} \end{pmatrix} \tag{2}$$

3. Scheme Design for Covert Communication Based on Matrix Decomposition Method of Digital Currency Transaction Amount

In this scheme, the sender Alice sends a message M to the receiver Bob, and Bob receives the message M. It needs to go through three stages, which are message encryption, message transmission, and message restoration. As shown in **Fig. 1**.

Message encryption stage: Firstly, the message M is encrypted and padded to generate M_C. Secondly, M_C is binary encoded to obtain M_B. Thirdly, M_B is cut, and the individual parts of the cut are converted to decimal numbers. Finally, the decimal numbers of each part are formed into a matrix $Matrix_{Number}$, which is decomposed into two transaction amount matrices $Matrix_a$ and $Matrix_b$.

Message transmission stage: Firstly, the set of transaction addresses generated by the pre-shared Key chain off the chain. Secondly, each column of matrix $Matrix_a$ consists of one transaction and each row of matrix $Matrix_b$ consists of one transaction. The transaction utilizes the interaction of the addresses. Finally, these transactions are published to the Bitcoin transaction network.

Message reduction stage: Firstly, generate the transaction address set according to the pre-shared Key off the chain, and extract the transactions. Secondly, the transaction amounts of these transactions are arranged separately to get two matrices $Matrix_a$ and $Matrix_b$, which are multiplied to get $Matrix_{Number}$. Thirdly, the elements Number on the diagonal of matrix $Matrix_{Number}$ are extracted and converted to binary code in turn to form M_B, which is decoded to get ciphertext M_C. Finally, the message M is obtained by decrypting the ciphertext M_C.

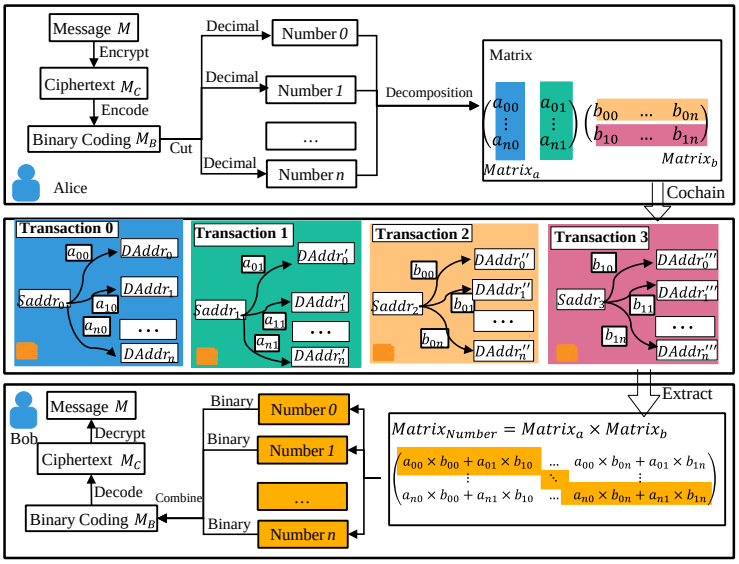


Fig. 1. Framework of the scheme based on the matrix decomposition method of digital currency transaction amounts to achieve covert communication

3.1 Message encryption stage

The message encryption stage includes the padding of the ciphertext, the cutting of the ciphertext, and the matrix decomposition.

3.1.1 Ciphertext padding and cutting

Ciphertext padding and cutting includes scheme design and algorithm design.

The design of ciphertext padding and cutting consists of padding the ciphertext M_C and cutting of ciphertext M_B. The purpose is to determine the size C_{pad} to be padded for M_C and the slice size $B_{cutCount}$ for M_B.

The ciphertext M_C is padded to make it easier to cut M_B,that is, the ciphertext M_C is padded to a certain length and encoded as M_B to make it easier to cut. Due to the cut of M_B, the last cut part will have less than the $B_{cutCount}$ bit binary length, it is difficult to decompose and needs to be padded to get M_C. The padding length is calculated as C_{pad} according to (3).

$$C_{pad} = (B_{cutCount}/8) - Ccount\%(B_{cutCount}/8) \tag{3}$$

The cut of the ciphertext M_B is the size of the slice to be obtained ($B_{cutCount}$). Firstly, the range $[min, max]$ is determined by the transaction amount length range d, according to (4) and (5). Secondly, max and min are converted to binary numbers respectively, and their bit numbers B_{max} and B_{min} are calculated according to (6) and (7). Thirdly, the number of bits in the slice must be a multiple of 8, since the conversion of a cipher character to binary requires 8 bits of binary to be represented. In order to make the embedding rate as large as possible within the specified range, the number of bits in the slice must be as large as possible, so $B_{cutCount}$ takes the maximum value, as shown in (8). Finally, in order to distribute the length of the transaction amount of the matrix decomposition in the most appropriate range, the range of values $[Number_{min}, Number_{max}]$ of the elements in the matrix $Matrix_{Number}$ is determined according to (9) and (10).

$$max = \underbrace{999\ldots999}_{d} \times \underbrace{999\ldots999}_{d} + \underbrace{999\ldots999}_{d} \times \underbrace{999\ldots999}_{d} \tag{4}$$

$$min = \underbrace{100\ldots000}_{d} \times \underbrace{100\ldots000}_{d} + \underbrace{100\ldots000}_{d} \times \underbrace{100\ldots000}_{d} \tag{5}$$

$$B_{max} = BinaryCount(max) \tag{6}$$

$$B_{min} = BinaryCount(min) \tag{7}$$

$$B_{cutCount} = Max(\{B_{cutCount} | B_{min} < B_{cutCount} < B_{max}\} \cap \{B_{cutCount} | B_{cutCount}\%8 = 0\}) \tag{8}$$

$$Number_{max} = Binary2decimal(\underbrace{111\ldots111}_{B_{cutCount}}) \tag{9}$$

$$Number_{min} = Binary2decimal(\underbrace{100\ldots000}_{B_{cutCount}}) \tag{10}$$

According to the analysis of the transaction amount length from the experiments in Section 4.2, it is most appropriate to set the transaction amount length d between 5 and 8. The above equation results in $B_{max} = 55$, $B_{min} = 28$ and $B_{cutCount}$ is taken as 48 bits for $d \in [5,8]$. $Number_{max}$=281474976710655, $Number_{min}$ =140737488355328.

In the ciphertext padding and cutting algorithm, firstly, after getting the message M and then encrypt M. As a result of cutting M_B, the last partial binary is less than 48, making it difficult to continue the decomposition, so padding is required to obtain M_C. Secondly, binary encoding of M_C to get M_B. Finally, cut M_B to convert each part of the cut to decimal number, get matrix $Matrix_{Number}$. The ciphertext padding and cutting algorithm is shown in Algorithm 1.

***Algorithm* 1**: Ciphertext padding and cutting algorithm
Input: M
Output: $Matrix_{Number}$

1)$M_C \leftarrow Encrypt(M)$;
2)$Ccount \leftarrow countString(M_C)$; // count the number of characters in M_C
3)**if** $Ccount\%6 \neq 0$
4) $C_{pad} \leftarrow 6 - Ccount\%6$;
5) **for** $i \leftarrow 1: C_{pad}$ **do**
6) $M_C \leftarrow RandomInsert('*')$; // Insert a random character '*' in M_C,to padding
7) **end for**
8)**end if**
9)$M_B \leftarrow StringtoBinary(M_C)$;// M_C converted to binary
10)**for** $i \leftarrow 0: countString(M_B)/B_{cutCount}$ **do**
11) $Number_i \leftarrow BinarytoDecimal(M_B[i * B_{cutCount}, (i+1) * B_{cutCount}])$;
12)**end for**
13)**return** $Matrix_{Number}$;

As in **Fig. 2**, Alice sends the message M("This is a secure channel for sending messages"), encrypted padding is performed, and then binary encoding is performed to obtain M_B="01010100011010000.....". Finally, a cut is performed and each part is decimal converted to $\{Number_0, Number_1, \cdots, Number_n\}$.

3.1.2 Novel matrix decomposition method

This section introduces the design of novel matrix decomposition methods and algorithm design, respectively.

The purpose of the matrix decomposition method designed in this scheme is to decompose the large amount (integer) matrix $Matrix_{Number}$ into two small amount matrices $Matrix_a$, $Matrix_b$, as shown in (1) and (2). The core concept is to decompose the diagonal element $Number_i$ of the large amount (integer) matrix $Matrix_{Number}$ into a composite number or two composite numbers (amount $\in N$). As shown in Corollary 1.

Corollary 1: Every natural number greater than 11 is the sum of two composite numbers [28] (a composite number is a number of natural numbers that is divisible by other numbers (except 0) in addition to 1 and itself).

1)If $n = 3k(k \geq 4)$,then

$$n = 3k = 6 + 3(k-2);$$

2)If $n = 3k + 1(k \geq 4)$,then

$$n = 3k + 1 = 4 + 3(k-1);$$

3)If $n = 3k + 2(k \geq 4)$,then

$$n = 3k + 2 = 8 + 3(k-2);$$

In any case, n can be expressed as the sum of two composite numbers.

Therefore, large amounts (integers) can be decomposed into one composite (the other composite is 0) or two composites, as shown in (11).

$$\{Number_i | Number_i = a_{i0} \times b_{0i} + a_{i1} \times b_{1i}, a_{i0} \in N, a_{i1} \in N, b_{0i} \in N, b_{1i} \in N\} \quad (11)$$

A natural number greater than 11 can be decomposed into two composite numbers, then the large amount (integer) matrix $Matrix_{Number}$ can be decomposed into two integer matrices $Matrix_a$, $Matrix_b$, as shown in Corollary 2.

Corollary 2: The large amount (integer) matrix $Matrix_{Number}$ can be decomposed into two integer matrices $Matrix_a$, $Matrix_b$.

1)If the factorization of $Number_i$ can be represented by a composite number, that is, $Number_i = a_{i0} \times b_{0i}$, a_{i1}=0, b_{1i}=0.

$$a_{i0} = \prod_{j=0}^{k} FactorNumber_j, b_{0i} = \prod_{j=k+1}^{m} FactorNumber_j$$

$$Number_i = (a_{i0} \quad 0) \times \begin{pmatrix} b_{0i} \\ 0 \end{pmatrix}$$

where k allows the length of a_{i0} and b_{0i} to be distributed in a prescribed range , and $FactorNumber_j$ denotes a factor of $Number_i$.

2) If $Number_i$ factorization can be represented by two composite numbers, that is, $Number_i = R1 + R2 = a_{i0} \times b_{0i} + a_{i1} \times b_{1i}$.

$$a_{i0} = \prod_{j=0}^{k1} R1factor_j \, , b_{0i} = \prod_{j=k1+1}^{l} R1factor_j$$

$$a_{i1} = \prod_{j=0}^{k2} R2factor_j \, , b_{1i} = \prod_{j=k2+1}^{\mathrm{h}} R2factor_j$$

$$Number_i = (a_{i0} \quad a_{i1}) \times \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix}$$

where $k1$ and $k2$ are such that the length distributions of a_{i0} , b_{0i} and a_{i1} , b_{1i} are in a prescribed range, respectively, and $R1factor_j$ denotes the factor of $R1$ and $R2factor_j$ denotes the factor of $R2$.

Therefore, the large amount (integer) matrix $Matrix_{Number}$ can be decomposed into two integer matrices $Matrix_a$, $Matrix_b$, as shown in (1) and (2).

In the novel matrix decomposition algorithm, after obtaining the matrix $Matrix_{Number}$, the matrix is decomposed into two transaction amount matrices $Matrix_a$ and $Matrix_b$. The decomposition algorithm is shown in Algorithm 2.

***Algorithm* 2**: Novel matrix decomposition algorithm

Input: $Matrix_{Number} = \begin{pmatrix} Number_0 & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & Number_n \end{pmatrix}$

Output: $Matrix_a, Matrix_b$

for $i \leftarrow 0{:}\, n$ **do**
  $a_{i0} \leftarrow 1, b_{0i} \leftarrow 1$;
  $FactorNumber[m] \leftarrow Factorization(Number_i)$; //Obtain all factors of $Number_i$
  if $FactorNumber_{m-1} \le MaxAmount$ && $FactorNumber_{m-1} \ge MinAmount$ **do**
    $(a_{i0} \quad 0), \begin{pmatrix} b_{0i} \\ 0 \end{pmatrix} \leftarrow LargeNumberDecomposition1(FactorNumber[m])$;
  end if
  if $FactorNumber_{m-1} < MinAmount$
    $(a_{i0} \quad 0), \begin{pmatrix} b_{0i} \\ 0 \end{pmatrix} \leftarrow LargeNumberDecomposition2(FactorNumber[m], Number_i)$;
  end if
  if $FactorNumber_{m-1} > MaxAmount$
    $(a_{i0} \quad a_{i1}), \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} \leftarrow LargeNumberDecomposition3(FactorNumber[m])$;
  end if
end for
$Matrix_a = \begin{pmatrix} a_{00} & a_{01} \\ \vdots & \vdots \\ a_{n0} & a_{n1} \end{pmatrix}, Matrix_b = \begin{pmatrix} b_{00} & \cdots & b_{0n} \\ b_{10} & \cdots & b_{1n} \end{pmatrix}$;
return $Matrix_a, Matrix_b$;

In this algorithm, $Number_i (i = 0 \ldots n)$ in matrix $Matrix_{Number}$ is factorized into a set of factors $FactorNumber_i (i = 0, \ldots, m-1)$. In turn, we judge whether the largest factor $FactorNumber_{m-1}$ in the factorization of each number $Number_i (i = 0 \ldots n)$ is within the normal range of transaction amounts $[MinAmount, MaxAmount]$, and there are three situations.

If $FactorNumber_{m-1}$ is within the normal range of transaction amount, that is, $FactorNumber_{m-1} \in [MinAmount, MaxAmount]$. Then $a_0 = FactorNumber_{m-1}$, $a_1 =$

0, $b_0 = \prod_{j=0}^{m-2} FactorNumber_j$, $b_1 = 0$. As shown in Algorithm 3.

***Algorithm* 3**: LargeNumberDecomposition1 algorithm
Input: $FactorNumber[m]$
Output: $(a_0 \quad 0), \binom{b_0}{0}$

1)$a_0 \leftarrow FactorNumber_{m-1}$;
2)$b_0 \leftarrow \prod_{j=0}^{m-2} FactorNumber_j$;
3)**return** $(a_0 \quad 0), \binom{b_0}{0}$;

***Algorithm* 4**: LargeNumberDecomposition2 algorithm
Input: $FactorNumber[m], Number$
Output: $(a_0 \quad 0), \binom{b_0}{0}$

1)**for** $k \leftarrow 0: m-1$ **do**
2) **if** $b_0 > MinAmount$
3) $b_0 \leftarrow b_0 \times FactorNumber_k$;
4) **end if**
5) $a_0 \leftarrow Number/b_0$;
6)**end for**
7)**return** $(a_0 \quad 0), \binom{b_0}{0}$;

If $FactorNumber_{m-1}$ is less than the minimum amount, that is, $FactorNumber_{m-1} < MinAmount$. then factor in $FactorNumber[m]$, multiply, and get $b_0 \in [MinAmount, MaxAmount]$, then record it, $a_0 = Number/b_0$, as shown in Algorithm 4.

If $FactorNumber_{m-1}$ is greater than the maximum transaction amount, that is, $FactorNumber_{m-1} > MaxAmount$. Therefore, $FactorNumber_{m-1}$ needs to be decomposed, as shown in Algorithm 5. Firstly, use A to record the product of factors other than the largest factorization, that is, $A \leftarrow \prod_{j=0}^{m-2} FactorNumber_j$. Then decompose $FactorNumber_{m-1}$ into two numbers $R_1 = FactorNumber_{m-1}/2$ and $R_2 = FactorNumber_{m-1} - R_1$. Secondly, factorize these two numbers R_1, R_2 respectively, and the factor sets are $R1factor_l$, $R2factor_h$,with b_0 and b_1 to record A, that is, $b_0 = A$, $b_1 = A$. Thirdly, multiply the numbers from $R1factor_l$ factor set with b_0 so that $b_0 > MinAmount$, multiply the numbers from $R2factor_h$ factor set with $b_1 > MinAmount$, and then calculate a_0 , a_1 , that is, $a_0 = (R_1 \times A)/\, b_0$, $a_1 = (R_2 \times A)/\, b_1$. Finally, determine whether a_0, a_1, b_0, b_1 all satisfy the transaction amount range, if so will get a_0, a_1, b_0, b_1. otherwise will be randomly regenerated (that is, $R_1 \leftarrow R_1 + Random(sqrt(R_1))$), and then decomposed until a_0, a_1, b_0, b_1 satisfy the transaction amount range.

***Algorithm* 5**: LargeNumberDecomposition3 algorithm
Input: $FactorNumber[m]$
Output: $(a_0 \quad a_1), \binom{b_0}{b_1}$

1)$A \leftarrow \prod_{j=0}^{m-2} FactorNumber_j$;
2)$R_1 \leftarrow FactorNumber_{m-1}/2$;
3)**while**$(true)$
4) $R_2 \leftarrow FactorNumber_{m-1} - R_1$;
5) $R1factor_l \leftarrow Factorization(R_1), R2factor_h \leftarrow Factorization(R_2)$;
6) $b_0 \leftarrow A, b_1 \leftarrow A$;
7) **while** $(b_0 < MinAmount \;\&\&\; k < l)$
8) $b_0 \leftarrow b_0 \times R1factor_k, k++$;
9) **end while**
10) **while** $(b_1 < MinAmount \;\&\&\; k < h)$
11) $b_1 \leftarrow b_1 \times R2factor_k, k++$;
12) **end while**
13) $a_0 \leftarrow \frac{R_1 \times A}{b_0}, a_1 \leftarrow \frac{R_2 \times A}{b_1}$;
14) **if** $Judge(a_0, a_1, b_0, b_1)$//Judge() is to determine whether the amount meets the amount range $[MinAmount, MaxAmount]$
15) $break$;
16) **end if**
17) $R_1 \leftarrow R_1 + Random(sqrt(R_1))$;// Generate R_1 randomly within the prescribed range
18)**end while**
19)**return** $(a_0 \quad a_1), \binom{b_0}{b_1}$;

In the novel matrix decomposition algorithm 2, the number $Number_i(i = 0, \ldots, n)$ in the matrix is finally decomposed into the sum of two composite numbers, as shown in (12)-(15).

$$Number_0 = a_{00} \times b_{00} + a_{01} \times b_{10} \tag{12}$$

$$Number_1 = a_{10} \times b_{01} + a_{11} \times b_{11} \tag{13}$$

$$\cdots$$

$$Number_n = a_{n0} \times b_{0n} + a_{n1} \times b_{1n} \tag{14}$$

$$\begin{pmatrix} Number_0 & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & Number_n \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ \vdots & \vdots \\ a_{n0} & a_{n1} \end{pmatrix} \times \begin{pmatrix} b_{00} & \cdots & b_{0n} \\ b_{10} & \cdots & b_{1n} \end{pmatrix}$$

$$= \begin{pmatrix} a_{00} \times b_{00} + a_{01} \times b_{10} & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & a_{n0} \times b_{0n} + a_{n1} \times b_{1n} \end{pmatrix} \tag{15}$$

The set of Number obtained by cutting in **Fig. 2** is decomposed into the amount matrix $Matrix_a$ and $Matrix_b$ by using matrix decomposition algorithm, as shown in **Fig. 3** Matrix decomposition example figure.

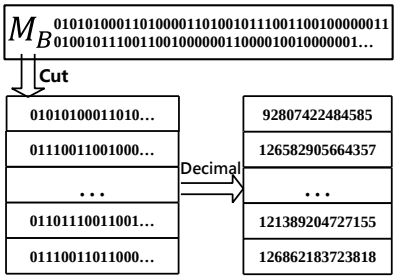


Fig. 2. Example figure of ciphertext padding and cutting

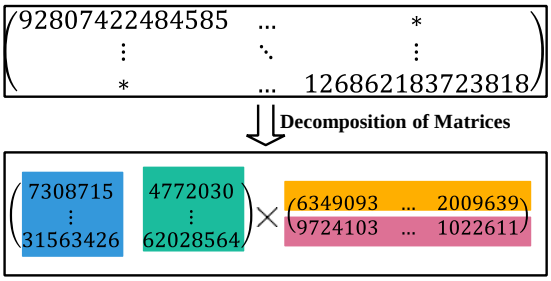


Fig. 3. Matrix decomposition example

3.2 Message transmission stage

In the message transmission stage, it is the posting of the transaction to the Bitcoin transaction network. Firstly, the design of the transaction address generation and the design of the interactions for each transaction, secondly, the transactions are carried out based on the transaction amounts in the two transaction matrices, and finally, these transactions are posted to the Bitcoin transaction network.

3.2.1 Transaction address generation design

With the pre-shared Key off the chain, we generate the sender's address set $Saddr$ and receiver's address set $DAddr$ for each transaction. Then, through $DAddr$, we further generate the receiver address set $DAddr'$, $DAddr''$ and $DAddr'''$ for each transaction. These addresses form a chain relationship with each other, which can be seen in **Fig. 4**.

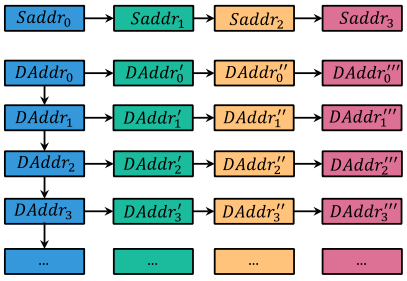


Fig. 4. Address chain relationship

Firstly, we generate the sender address set $Saddr$ for these four transactions and the $DAddr$ address set in the first transaction using Key chaining, as detailed in Algorithm 6. Subsequently, through the $DAddr$ set, we derive the $DAddr'$, $DAddr''$, $DAddr'''$ address set ($DAddr \rightarrow DAddr' \rightarrow DAddr'' \rightarrow DAddr'''$), with the specific address derivation relation $addr_{sk(i+1)} \leftarrow addr_{sk(i)} * G * T$, as detailed in Algorithm 7. The primary purpose of Algorithm 6 and Algorithm 7 is to provide for the receiver to locate the transaction and the interaction of the transaction amount between addresses in order to efficiently perform subsequent processing.

***Algorithm* 6:** $Saddr$ and $DAddr$ address sets generation algorithm
Input: Key
Output: $Saddr, DAddr$

$Saddr_{sk(0)} \leftarrow Hex(Key * 4)$;
$Saddr_{pk(0)} \leftarrow SECP256K1(Saddr_{sk(0)})$; //$SECP256K1()$ is the elliptic curve equation
$Saddr_0 \leftarrow Base58\left(Ripemd160\left(Sha256\left(Saddr_{pk(0)}\right)\right)\right)$;
for $i \leftarrow 0:3$ **do**
    $Saddr_{sk(i+1)} \leftarrow Saddr_{sk(i)} * G$;
    $Saddr_{pk(i+1)} \leftarrow SECP256K1(Saddr_{sk(i+1)})$;
    $Saddr_{(i+1)} \leftarrow Base58\left(Ripemd160\left(Sha256\left(Saddr_{pk(i+1)}\right)\right)\right)$;
end for
$DAddr_{sk(0)} \leftarrow Hex(Key * 2)$;
$DAddr_{pk(0)} \leftarrow SECP256K1(DAddr_{sk(0)})$;
$DAddr_{(0)} \leftarrow Base58\left(Ripemd160\left(Sha256\left(DAddr_{pk(0)}\right)\right)\right)$;
for $i \leftarrow 0:n$ **do**
    $DAddr_{sk(i+1)} \leftarrow DAddr_{sk(i)} * G$;
    $DAddr_{pk(i+1)} \leftarrow SECP256K1(DAddr_{sk(i+1)})$;
    $DAddr_{(i+1)} \leftarrow Base58\left(Ripemd160\left(Sha256\left(DAddr_{pk(i+1)}\right)\right)\right)$;
end for
$Saddr \leftarrow \{Saddr_0, Saddr_1, Saddr_2, Saddr_3\}$;
$DAddr \leftarrow \{DAddr_0, DAddr_1, \cdots, DAddr_n\}$;
return $Saddr, DAddr$;

***Algorithm* 7**: $DAddr^*$ address sets generation algorithm
Input: $DAddr$
Output: $DAddr', DAddr'', DAddr'''$

$DAddr^{(0)} \leftarrow DAddr'$
for $i \leftarrow 0:3$ **do**
    $DAddr_{sk}{}^{(i+1)} \leftarrow DAddr_{sk}{}^{i} * G * 4$
    $DAddr_{pk}{}^{(i+1)} \leftarrow SECP256K1(DAddr_{sk}{}^{(i+1)})$;
    $DAddr^{(i+1)} \leftarrow Base58\left(Ripemd160\left(Sha256\left(DAddr_{pk}{}^{(i+1)}\right)\right)\right)$;
end for
$DAddr' \leftarrow DAddr^{(1)}$ □ $DAddr'' \leftarrow DAddr^{(2)}$ □ $DAddr''' \leftarrow DAddr^{(3)}$;
return $DAddr', DAddr'', DAddr'''$;

3.2.2 Design of transaction amount and address interaction

According to the transaction amount matrix $Matrix_a$, $Matrix_b$ and address generation relationship, the corresponding amount is used as the transaction amount for each transaction. The first column in $Matrix_a$ represents the transactions in Transaction0, where the relationship is ($Saddr_0 \xrightarrow{a_{i0}} DAddr_i$), and the second column represents the transactions in Transaction1, where the relationship is ($Saddr_1 \xrightarrow{a_{i1}} DAddr_i'$).The first row in $Matrix_b$ represents the transactions in Transaction2, where the relationship is ($Saddr_2 \xrightarrow{b_{0i}} DAddr_i''$), and the second row represents the transactions in Transaction3, where the relationship is

($Saddr_3 \xrightarrow{b_{1i}} DAddr_i'''$). The transaction address interaction is shown in **Fig. 5**. Finally post these four transactions to the Bitcoin transaction network.

According to the transaction matrix $Matrix_a$, $Matrix_b$ obtained from **Fig. 3**, its transaction matrix is assigned to 4 transactions as shown in **Fig. 6**. Because the unit of numbers in $Matrix_a$, $Matrix_b$ is Satoshi and the unit in Bitcoin transaction network is BTC, 1BTC=10^8Satoshi, the size of transaction matrix $Matrix_a$, $Matrix_b$ is reduced by 10^8 times.

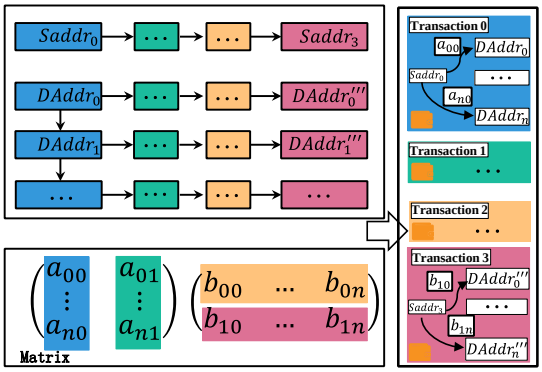


Fig. 5. Transaction Address Interaction Relationship

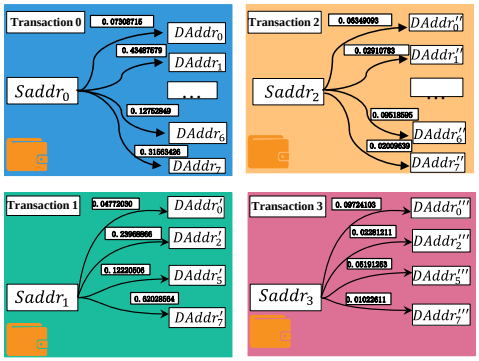


Fig. 6. Examples of transactions

3.3 Message reduction stage

The receiver can derive the receiver address set $Saddr$ for each transaction by using the pre-shared Key off-chain and using the address set generation algorithm. Find these 4 transactions and get the transaction amount matrix $Matrix_a$, $Matrix_b$, then $Matrix_{Number} = Matrix_a \times Matrix_b$. The algorithm of reducing message is shown in Algorithm 8.

***Algorithm* 8**: Reduction message algorithm
Input: $Matrix_a, Matrix_b$
Output: M

1)**for** $i \leftarrow 0:n$ **do**
2) $Number_i \leftarrow a_{i0} \times a_{i1} + b_{0i} \times b_{1i}$;
3) $BinaryNumber \leftarrow DecimaltoBinary(Number_i)$; // Decimal to Binary
4) **if** $count(BinaryNumber) < B_{cutCount}$
5) $BinaryNumber \leftarrow '0' * (B_{cutCount} - count(BinaryNumber)) + BinaryNumber$;
6) **end if**
7) $M_B \leftarrow M_B + BinaryNumber$;
8)**end for**
9)$M_C \leftarrow Decode(M_B)$;
10)$M_C \leftarrow M_C.Remove('*')$;// Remove character $'*'$ from M_C
11)$M \leftarrow Decrypt(M_C)$;
12)**return** M;

Firstly, determine the transaction according to the address in $Saddr$, determine the transaction, and then extract the transaction amount from each transaction to form the matrices $Matrix_a$, $Matrix_b$. Secondly, the two matrices are multiplied to obtain the matrix $Matrix_{Number}$. Thirdly, the diagonal numbers in this matrix will be converted, filled and connected in order to finally get the binary code M_B. Finally, decode it to get M_C, remove the padding character $'*'$ from M_C, and then decrypt it to get the message M.

4. Experiments and Analysis

In this section, firstly, different matrix decomposition methods are compared and analyzed in four aspects, secondly, the scheme is analyzed in terms of resistance to detection, thirdly, the embedding rate comparison is analyzed, and finally, the performance of this scheme is compared with other schemes. The experimental environment for this article: Windows 11, CPU is AMD Ryzen 7 5800H with Radeon Graphics, main frequency is 3.20GHz, 16G RAM.

4.1 Comparative analysis of different matrix decomposition methods

In this subsection, the feasibility of the matrix decomposition method of this scheme is compared and analyzed from four aspects.

4.1.1 Comparative analysis of the correctness of different matrix decomposition methods

To verify the correctness of the matrix decomposition method is to decompose the matrix $Matrix_{Number}$ by using the matrix decomposition method, and verify whether the result of the decomposition conforms to the prescribed length range d. where the elements of matrix $Matrix_{Number}$ appear in the range $[Number_{max}, Number_{min}]$, and it is derived in Section 3.1. $Number_{max} = 281474976710655$, $Number_{min} = 140737488355328$.

In this range $[Number_{max}, Number_{min}]$, some numbers are randomly selected to form matrices $Matrix_{Number}$ of different orders, the matrices $Matrix_{Number}$ are decomposed using different matrix decomposition methods, and the decomposed matrices are observed. The Triangular decomposition method decomposes a matrix into a lower triangular matrix (L) and an upper triangular matrix (U), whose decomposed matrix has elements whose lengths are not in the range of d and which also contain decimals. Therefore, the result of the decomposition is “False”. The Full-Rank decomposition method decomposes a matrix into two matrices, and the lengths of the elements of the decomposed matrix do not match the length range. Therefore, the result of the decomposition is “False”. The QR decomposition method decomposes a matrix into an orthogonal matrix (Q) and an upper triangular matrix (R), and the lengths of the elements of the decomposed matrix do not match the length range. Therefore, the result of the decomposition is “False”. The Singular Value decomposition method (SVD) decomposes a matrix into column orthogonal matrices (U), diagonal matrices (Σ), and the transpose V^T of a positive definite matrix, and the lengths of the elements in the decomposed matrix do not fit into the length range. Therefore, the result of the decomposition is “False”. The matrix decomposition method of this program decomposes the matrix into two matrices, and the lengths of the elements in the matrix after its decomposition are in the length range. Therefore, the result of the decomposition is “Ture”. The results of the various matrix decomposition methods are shown in **Table 1**. If the result of the decomposition matches the number of its prescribed length, then it is “Ture”, otherwise it is “False”.

Table 1. Comparison of the correctness of different matrix decomposition methods

Order of matrix	Triangular decomposition	Full Rank decomposition	QR decomposition	Singular Value decomposition(SVD)	The matrix decomposition method of this scheme
1	False				Ture
…					
100					

In **Table 1**, it can be seen that the results of Triangular decomposition, Full Rank decomposition, QR decomposition and SVD (singular value) decomposition for matrices of this range are not able to satisfy the prescribed length of compliance. The matrix decomposition method designed in this scheme can satisfy that the decomposition results conform to the prescribed length. Therefore, the matrix decomposition method of this scheme is feasible.

4.1.2 Comparative analysis of transaction amount consumption of different matrix decomposition methods

Comparative analysis of transaction amount consumption for different matrix decomposition methods is to analyze the elements in the matrix after decomposing the amount matrices $Matrix_{Number}$ produced by messages of different lengths using various matrix decomposition methods. Where transaction amount consumption is the sum of the elements in the decomposed matrix. The methods compared include Triangular matrix decomposition, Full Rank matrix decomposition, QR matrix decomposition, SVD matrix decomposition and traditional methods.

Through simulation experiments, the sum of the elements of the matrices produced by different methods for messages of different lengths is recorded. The different length information is manipulated to make the conversion to an amount matrix $Matrix_{Number}$. Various library functions for matrix decomposition in the compiler and the matrix decomposition method of this scheme are called to perform the decomposition and the sum of the elements in the resulting matrix after the decomposition is recorded. As the transaction amount cannot be negative, Triangular matrix decomposition, Full Rank matrix decomposition, QR matrix decomposition and SVD matrix decomposition will produce negative numbers, so it is necessary to process the absolute value of the elements in the factorized matrix before summing. And in the comparative analysis, the traditional method is added for comparison, where the traditional method is to transmit the covert message directly in amounts without matrix decomposition. Therefore, the traditional method records the sum of the numbers in the matrix to be decomposed. To avoid experimental contingency, we conducted 1000 experiments on the amounts generated by messages of different lengths, and the results were averaged and recorded. The results of the amount consumption comparison are shown in **Fig. 7**.

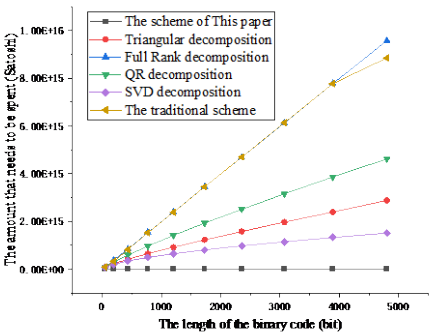


Fig. 7. Different matrix decomposition of transaction amount consumption

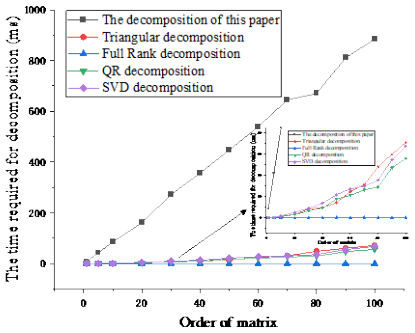


Fig. 8. Comparison of decomposition efficiency of different matrix decomposition methods

In **Fig. 7**, the matrix decomposition method of this scheme consumes much less amount at different lengths than the other methods. The matrix decomposition method of this scheme increases the consumption of transaction amount slowly as the length of message increases.

The traditional and Full Rank decomposition methods consume similar amounts of transactions for different message lengths. The traditional method is to trade directly with amounts without decomposing the matrix, and what is recorded is the sum of the elements in the matrix to be decomposed. Although the Full Rank decomposition method decomposes the matrix, the decomposed matrix is almost the same as the decomposed one. This results in Full Rank decomposition methods that require similar amounts of consumption as traditional methods. Therefore, the matrix decomposition method of this scheme greatly reduces the consumption of the amount.

4.1.3 Comparative analysis of decomposition efficiency of different matrix decomposition methods

The matrix decomposition efficiency is the time taken to factorize a matrix. This section is to decompose matrices of different orders and record the time required for different methods of matrix decomposition respectively. The random numbers are drawn from the range $[Number_{max}, Number_{min}]$ to form matrices $Matrix_{Number}$ of different orders. And call various library functions for matrix decomposition in the compiler and the matrix decomposition method of this scheme to perform the decomposition and record the time consumed. Meanwhile, in order to avoid the contingency of the experiment, each order matrix is randomly generated 1000 times, and then the average value is recorded. The experimental results are shown in **Fig. 8**.

In **Fig. 8**, it can be seen that the time required to decompose the matrix in this scheme is the most consumed. The reason is that since the matrix decomposition method of this scheme decomposes each element on the diagonal of the matrix into one composite number or two composite numbers, its decomposition process and verification process will be more time consuming. And the other matrix decomposition methods are decomposing the matrix directly without the limitation of length and decimals, which makes the other matrix decomposition methods more efficient. In **Fig. 8**, although the efficiency of the matrix decomposition method of this scheme is much lower than other matrix decomposition methods, the matrix decomposition method of this scheme takes only 883.9698482ms, less than 1s, for a matrix of order 100. And it can be seen from **Fig. 7** that the amount consumed by other matrix decomposition methods is much higher than the amount consumed by the present scheme. Therefore, the matrix decomposition method of the present scheme is desirable.

4.1.4 Comparative analysis of embedding rate of different matrix decomposition methods

The embedding ratio is the ratio between the length of a message and the number of transactions required to transmit a message of that length. The comparative analysis of embedding rate for different matrix decomposition methods is to analyze the comparison of embedding rate produced by different lengths of messages using different matrix decomposition methods. Firstly, encrypt and encode the messages of different lengths to be processed to obtain the large amount matrix $Matrix_{Number}$. Secondly, the large amount matrix is decomposed by calling various library functions for matrix decomposition in the compiler and the matrix decomposition method of this scheme, and the number of non-zero elements in the decomposed matrix, which is the number of transactions needed, is recorded. To avoid the contingency of the experiment, messages of different lengths are randomly generated 1000 times and decomposed. Finally, the number of non-zero elements in the matrix generated after decomposition by different matrix decomposition methods is counted, averaged and recorded. The experiments are compared and analyzed as shown in **Fig. 9**.

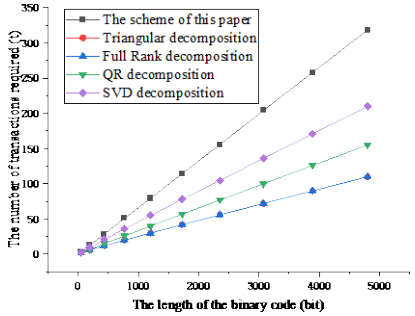


Fig. 9. Comparison of embedding rates of different matrix decomposition methods

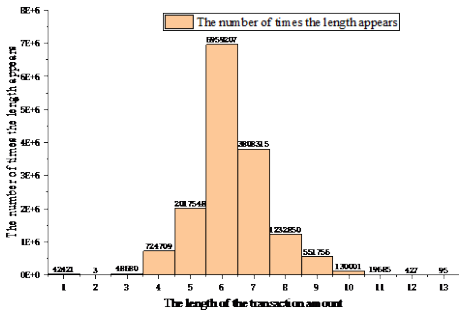


Fig. 10. Distribution of transaction amount length

In **Fig. 9**, the embedding rate of this scheme is lower than other matrix decomposition methods. The Triangular decomposition method and the Full Rank decomposition method appear to overlap in **Fig. 9**, which is due to the large elements in the decomposed matrix. The Triangular decomposition method decomposes the matrix into an upper triangular matrix and a lower triangular matrix, which are all non-zero elements. The Full Rank Decomposition method decomposes the matrix into two matrices, the first matrix with all non-zero elements and the second matrix with all non-zero elements on the diagonal only. As a result, it leads to equal sum of non-zero elements in the matrix after the Triangular decomposition method and the Full rank decomposition method. Although the other matrix decomposition methods have high embedding rates, it can be seen from **Table 1** that the other matrix decomposition methods cannot effectively control the elements in the decomposed matrix, and there are problems such as fractional and minus numbers in the decomposed matrix. Therefore, the matrix decomposition method of this scheme is the most suitable one to be chosen.

4.2 Resistance to detection

The transaction has problems such as easy detection if it contains attributes such as special transaction amounts (e.g., specially large, specially small) [29][30]. Since the covert communication carrier in the design of this scheme is the transaction amount, it is important to control the transaction amount so that the transaction carrying the information has some resistance to detection. In order to avoid that this scheme will generate special transaction amounts, statistical analysis of the amount lengths in real transaction networks and analysis of the frequency of transaction amounts are needed to obtain the range of amount lengths d designed in this scheme.

Firstly, the statistical analysis of the amount lengths in the real transaction network is performed to determine the real range of transaction amount lengths. Since this scheme is implemented on the Bitcoin transaction network, we crawled the latest transaction blocks (769473-771632) from the Bitcoin transaction network for statistical analysis, and crawled a total of 15535697 transaction amounts. Crawl the transaction amount in the transaction block in Satoshi units (1BTC=10^8Satoshi) and count the number of occurrences of the length of each transaction amount. As shown in **Fig. 10**.

Secondly, to further determine the range of lengths d, the frequency of amounts in each length was counted. We refer to the detection resistance experiments in the literature [21], whose experiments determine the transaction amount length from the perspective of fluctuations through the transaction amount and make the NCCM scheme resistant to detection. In order to avoid the impact of special transaction amounts on the statistical experiment of transaction amount frequency, and also to ensure the number of experimental data sets. The

crawled amount dataset is first arranged in positive order, the small amount data in the first 5% of the sorted amount dataset and the large amount data in the last 5% are deleted, and the frequency statistics are analyzed for the middle 90% of the data. Where the middle 90% of the data, the range of the amount length can be calculated as [4,8] according to **Fig. 10**. Then, we verify that the amounts are uniformly distributed in each length, and **Fig. 11** shows the frequency distribution of transaction amounts. The frequency statistics for the amount range [0,1.0E+8], with 2.0E+7 as the step size, are shown in **Fig. 11 (a)**. Although the amount of the range [0,2.0E+7] accounts for 95.352%, the amount of the interval [2.0E+7, 1.0E+8] is more uniform. In order to verify whether the amount data of length 8 are uniformly distributed, that is, to verify whether the amounts in the range [1.0E+7,1.0E+8] are uniformly distributed, this experiment performs a frequency analysis for the range [0,2.0E+7], as shown in **Fig. 11 (b)**. In **Fig. 11 (b)**, the range [1.0E+7,2.0E+7] accounts for 3.811% of the range [0,2.0E+7], therefore, the amount length is 8 when the amount of the range [1.0E+7,1.0E+8]. The frequency distribution of this range is more uniform and the length can be 8. In **Fig. 11 (b)**, the frequency of the amount range [0,2.0E+6] is 79.985%, and the amount of the interval [2.0E+6,1.0E+7] is uniformly distributed. To verify whether the amounts of length 7 are evenly distributed, that is, to verify whether the amounts of the range [1.0E+7,1.0E+8] are uniformly distributed, this experiment performs frequency statistics for the amount range [0,2.0E+6] with 2.0E+5 as the step size, as shown in **Fig. 11 (c)**. It can be seen from **Fig. 11 (b)** and **Fig. 11 (c)** that the length is 7 when the data of the range [1.0E+6,1.0E+7]. The frequency distribution of this range is uniform and does not fluctuate much, therefore, the length can also be 7. For the amount range [0, 2.0E+5], frequency statistics were performed with 2.0E+4 as the step size, as shown in **Fig. 11 (d)**. For the amount range [0, 2.0E+4], frequency statistics are performed with 2.0E+3 as the step size, as shown in **Fig. 11 (e)**. Since the statistical range [0,8.0E+3] has a frequency of 0, this range is combined together in **Fig. 11 (e)**. From **Fig. 11 (c)**, **Fig. 11 (d)** and **Fig. 11 (e)**, it can be seen that the frequency of each range is more uniform when the length is 5 and 6, so the length can be 5 or 6. From **Fig. 11 (e)**, it can be seen that in the range [0,1.0E+4] only the range [8.0E+3,1.0E+4] has a frequency, and when the length is lower than 5, that is, the range [0, 1.0E+4], the data will be distributed in the first 5% of the small amount data with some specificity. Therefore, the length can be 5, but not lower than 5.

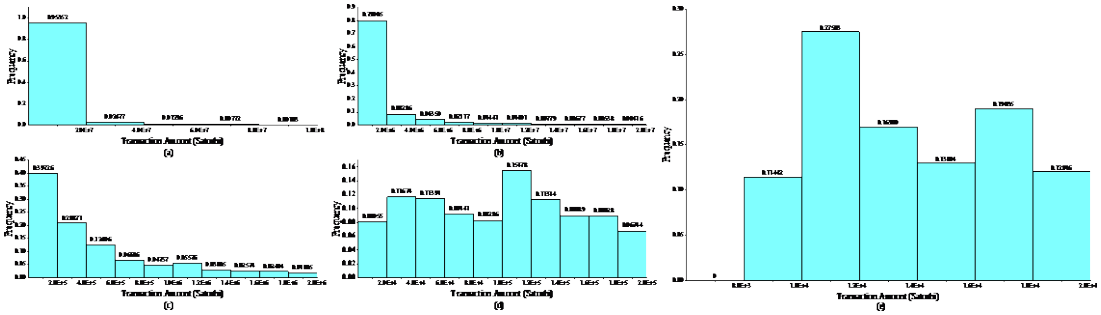


Fig. 11. Frequency distribution of transaction amount

Finally, in the method of designing matrix decomposition, the length of the amount of the decomposed amount matrix is controlled to be 5-8 ($d \in$[5,8]), so that the amount carrying covert messages has some resistance to detection.

4.3 Comparative analysis of embedding rate

The comparative embedding rate analysis is a comparative embedding rate analysis of this scheme with several amount-based covert communication schemes. To simulate the embedding rate of this scheme, this section continues to imitate the embedding rate experiment of this scheme in Section 4.1.4. The messages are processed into large amount matrices $Matrix_{Number}$, which are decomposed using the matrix decomposition method designed in this scheme, and the number of non-zero elements in the decomposed matrix, which is the number of transactions required, is recorded. In the literature [19], the embedding rate of HMAC-based MBE scheme is 14.07 bit/t (average) and the embedding rate of Hash-MBE scheme is 28.12 bit/t (average) . The embedding rate comparison is shown in **Fig. 12**.

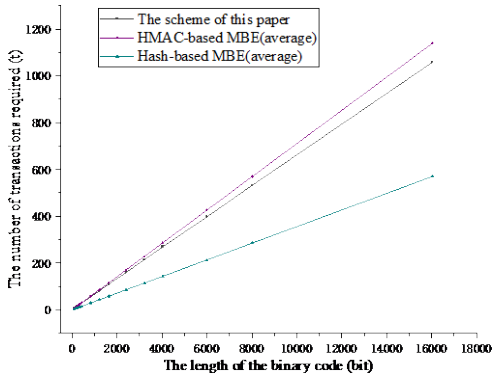


Fig. 12. Comparative analysis of embedding rate

From **Fig. 12**, it can be seen that the number of transactions required increases with the increase in the length of the binary encoded form ciphertext M_B. And the embedding rate of the scheme proposed in this paper fluctuates only slightly, and the average value of the embedding rate of the scheme can be calculated as 15.02 bit/t. At the same time, the proposed scheme is higher than HMAC-based MBE scheme and lower than Hash- MBE scheme compared with other schemes.

4.4 Performance comparison with other schemes

The performance comparison is done from two aspects with other schemes and its comparison is shown in **Table 2**. Firstly, this paper compares with other schemes in terms of embedding rate. The BLOCCE scheme[12] is to convert the information to binary embedded in the transaction address of each transaction with an embedding rate of 1 bit/t. The HMAC-based MBE scheme[19] is to embed the information onto the Value field of the Ethereum with an embedding rate of 14.07 bit/t. The Hash-based MBE scheme further improves on the HMAC-based MBE scheme by increasing the embedding rate, which is 28.12 bit/t. TISCC[22] on the chain is to embed the information is to the relevant field of the Ethereum transaction and call the smart contract to transmission the information with an embedding rate of 29 bit/t. In the scheme of this paper, the information is embedded to the transaction amount using a novel matrix decomposition method with an embedding rate of 15.02 bit/t. Finally, in terms of transmission efficiency, we compare how many transactions are required to complete the transmission of the message “This is a secret message” by different schemes. The BLOCCE scheme requires 192 transactions as each bit is embedded on each address. HMAC-based MBE and Hash-based MBE can calculate the number of transactions based on their embedding rates, which are 14 transactions and 7 transactions, respectively. NCCM[21] is a transmission of

information embedded in transaction amounts and interaction relationships, and the number of transactions required is 16 transactions. This scheme is experimentally simulated to require 12 transactions.

Table 2. Performance comparison with other schemes

	BLOCCE [12]	**HMAC-based MBE[19]**	**Hash-based MBE[19]**	**NCCM [21]**	**TISCC [22]**	**The scheme of this paper**
Embedding Rate	1 bit/t	14.07bit/t (average)	28.12 bit/t (average)	-	29 bit/t	15.02bit/t
Number of transactions required to transmit the message “This is a secret message”	192	14	7	16	-	12
Resistance to detection	-	Yes	Yes	Yes	-	Yes
Environment	Bitcoin	Ethereum	Ethereum	Bitcoin	Ethereum	Bitcoin

5. Conclusion

In this paper, we propose a covert communication scheme based on the matrix decomposition method of digital currency transaction amount, which effectively alleviates the problems of low embedding rate, large transaction amount and easy detection. The scheme employs a novel matrix decomposition method to improve the embedding rate and give the scheme some resistance to detection. However, the scheme may lead to greater amount consumption compared to schemes that do not involve amounts as carriers. Compared to the scheme designed with amounts as carriers, the scheme successfully reduces the consumption of transaction amounts and further improves the anti-detectability by skillfully distributing the decomposed transaction amount matrix elements in the most appropriate length range. The experimental results show that the scheme achieves an embedding rate of 15.02 bit/t within the optimal length range [5, 8]. Therefore, not only the embedding rate and the reduced transaction amount consumption are improved, but also the resistance to detection. In the future, we can apply the scheme in this paper to covert communication scenarios with numbers. For example, in financial transactions, covert communication with numbers can be used to confirm the authenticity of the transaction. By hiding some covert information in the amount or transaction code, both parties can verify the validity of the transaction and prevent fraud during the transaction process.

Acknowledgement

This work is sponsored by the National Natural Science Foundation of China No. 62172353, No. 62302114, No. U20B2046 and No. 61976064. Future Network Scientific Research Fund Project No. FNSRFP-2021-YB-48. Innovation Fund Program of the Engineering Research Center for Integration and Application of Digital Learning Technology of Ministry of Education No.1221045.

References

[1] B. W. Lampson, “A note on the confinement problem,” *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, 1973. Article (CrossRef Link)

[2] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods,” *Signal processing*, vol. 90, no. 3, pp. 727–752, 2010. Article (CrossRef Link)

[3] M. M. Sadek, A. S. Khalifa, and M. G. Mostafa, "Video steganography: a comprehensive review," *Multimedia tools and applications*, vol. 74, pp. 7063–7094, 2015. Article (CrossRef Link)
[4] M. R. Vinothkanna, "A secure steganography creation algorithm for multiple file formats," *Journal of Innovative Image Processing (JIIP)*, vol. 1, no. 01, pp. 20–30, 2019. Article (CrossRef Link)
[5] C. Zhao and Y. Guan, "A graph-based investigation of bitcoin transactions," in *Proc. of Advances in Digital Forensics XI: 11th IFIP WG 11.9 International Conference*, Orlando, FL, USA, pp. 79–95, 2015. Article (CrossRef Link)
[6] A. Sward, I. Vecna, and F. Stonedahl, "Data insertion in bitcoin's blockchain," *Ledger*, vol. 3, 2018. Article (CrossRef Link)
[7] A. J. Di Scala, A. Gangemi, G. Romeo, and G. Vernetti, "Special subsets of addresses for blockchains using the secp256k1 curve," *Mathematics*, vol. 10, no. 15, p. 2746, 2022. Article (CrossRef Link)
[8] S. Delgado-Segura, C. Pérez-Solà, J. Herrera- Joancomartí, G. NavarroArribas, and J. Borrell, "Cryptocurrency networks: A new p2p paradigm," *Mobile Information Systems*, vol. 2018, pp. 1–16, 2018. Article (CrossRef Link)
[9] T. Huynh-The, T. R. Gadekallu, W. Wang, G. Yenduri, P. Ranaweera, Q.-V. Pham, D. B. da Costa, and M. Liyanage, "Blockchain for the metaverse: A review," *Future Generation Computer Systems*, 2022. Article (CrossRef Link)
[10] J. Zhang, S. Zhong, T. Wang, H.-C. Chao, and J. Wang, "Blockchainbased systems and applications: a survey," *Journal of Internet Technology*, vol. 21, no. 1, pp. 1–14, 2020. Article (CrossRef Link)
[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, p. 21260, 2008. Article (CrossRef Link)
[12] J. Partala, "Provably secure covert communication on blockchain," *Cryptography*, vol. 2, no. 3, p. 18, 2018. Article (CrossRef Link)
[13] L. Zhang, Z. Zhang, W. Wang, R. Waqas, C. Zhao, S. Kim, and H. Chen, "A covert communication method using special bitcoin addresses generated by vanitygen," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 597-616, 2020. Article (CrossRef Link)
[14] L. Zhang, Z. Zhang, Z. Jin, Y. Su, and Z. Wang, "An approach of covert communication based on the ethereum whisper protocol in blockchain," *International Journal of Intelligent Systems*, vol. 36, no. 2, pp. 962–996, 2021. Article (CrossRef Link)
[15] W. Warren and A. Bandeali, "0x: An open protocol for decentralized exchange on the ethereum blockchain,". [Online]. Available: https://github.com/0xProject/whitepaper, pp. 04–18, 2017.
[16] J. Tian, G. Gou, C. Liu, Y. Chen, G. Xiong, and Z. Li, "Dlchain: A covert channel over blockchain based on dynamic labels," in *Proc. of Information and Communications Security: 21st International Conference, ICICS 2019*, pp. 814–830, 2020. Article (CrossRef Link)
[17] F. Gao, L. Zhu, K. Gai, C. Zhang, and S. Liu, "Achieving a covert channel over an open blockchain network," *IEEE Network*, vol. 34, no. 2, pp. 6–13, 2020. Article (CrossRef Link)
[18] A. V. Markelova, "Kleptographic (algorithmic) backdoors in the rsa key generator," *Prikladnaya Diskretnaya Matematika*, no. 55, pp. 14– 34, 2022. Article (CrossRef Link)
[19] S. Liu, Z. Fang, F. Gao, B. Koussainov, Z. Zhang, J. Liu, and L. Zhu, "Whispers on ethereum: Blockchain-based covert data embedding schemes," in *Proc. of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, pp. 171–179, 2020. Article (CrossRef Link)
[20] K. Jezek, "Ethereum data structures," *arXiv preprint arXiv:2108.05513*, 2021. Article (CrossRef Link)
[21] X. Luo, P. Zhang, M. Zhang, H. Li, and Q. Cheng, "A novel covert communication method based on bitcoin transaction," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2830–2839, 2022. Article (CrossRef Link)
[22] A. I. Basuki and D. Rosiyadi, "Joint transaction-image steganography for high capacity covert communication," in *Proc. of 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 41–46, 2019. Article (CrossRef Link)

[23] J. Yoo and S. Choi, "Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifolds," in *Proc. of Intelligent Data Engineering and Automated Learning–IDEAL 2008: 9th International Conference Daejeon*, pp. 140–147, 2008. Article (CrossRef Link)
[24] J. R. Bunch and J. E. Hopcroft, "Triangular factorization and inversion by fast matrix multiplication," *Mathematics of Computation*, vol. 28, no. 125, pp. 231–236, 1974. Article (CrossRef Link)
[25] R. Piziak and P. Odell, "Full rank factorization of matrices," *Mathematics magazine*, vol. 72, no. 3, pp. 193–201, 1999. Article (CrossRef Link)
[26] T. F. Chan, "Rank revealing qr factorizations," *Linear Algebra and its Applications*, vol. 88-89, pp. 67–82, 1987. Article (CrossRef Link)
[27] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM review*, vol. 35, no. 4, pp. 551–566, 1993. Article (CrossRef Link)
[28] L. M. Adleman, "On distinguishing prime numbers from composite numbers," in *Proc. of 21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*, IEEE, pp. 387–406, 1980. Article (CrossRef Link)
[29] Z. Gu, D. Lin, and J. Wu, "On-chain analysis-based detection of abnormal transaction amount on cryptocurrency exchanges," *Physica A: Statistical Mechanics and its Applications*, vol. 604, p. 127799, 2022. Article (CrossRef Link)
[30] L. Yang, X. Dong, S. Xing, J. Zheng, X. Gu, and X. Song, "An abnormal transaction detection mechanim on bitcoin," in *Proc. of 2019 International Conference on Networking and Network Applications (NaNA)*, pp. 452–457, 2019. Article (CrossRef Link)

Lejun Zhang, received his M.S. degree in computer science and technology in Harbin Institute of Technology and the Ph.D. degrees in computer science and technology at Harbin Engineering University, where he was an professor at Guangzhou University. His research interests include computer network, blockchain and information security.

Bo Zhang, is pursuing the M.E. degree in Electronic Information Engineering at the Yangzhou University, Yangzhou. His research interests include covert communication in blockchain and information security.

Ran Guo, received her B.S. degree and M.S. degree in Northeast Agricultural University. Where she works at Guangzhou University. Her research interests include computer network and artificial intelligence.

Zhujun Wang, received the B.Sc degree in Statistics from Yangzhou University, Yangzhou, China, in 2018, where he received his M.Sc degree in Software Engineering from Yangzhou University, in 2021. He is pursuing his Ph.D degree. His research interests include covert communication in blockchain, privacy protection, and blockchain consensus algorithms.

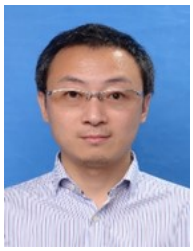
Guopeng Wang, Ph.D. in law, Jilin University. Works in Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education.

Jing Qiu, received the Ph.D. degree in computer applications technology from Beijing Institute of Technology. She is currently a Professor and Ph.D. Supervisor of the Cyberspace Institute of Advanced Technology, Guangzhou University. She was a Visiting Scholar with the University of Southern California, LA, USA, under the supervision of Professor Craig A. Knoblock. Her current research interest is Cyberspace Security, Knowledge Representation, and Big Data Analysis.

Shen Su is currently an associate professor at the Institute of Advanced Cyberspace Technology, Guangzhou University, and executive deputy director of Guangzhou University-Qianxin Cloud Security Joint Lab. He received his B.S., M.S., and Ph.D. from Harbin Institute of Technology and studied abroad at the University of Arizona from 2012-2013. His main research interests include blockchain security, DNS security, routing security, and vehicular network security.

Yuan Liu, PhD supervisor, is a young outstanding talent of Guangzhou University's "100 people plan", a member of CCF blockchain committee, and a member of China Communication Society blockchain committee. Her main research interests include blockchain consensus incentive mechanism, federated learning blockchain, data privacy security, network security offensive and defensive games, etc.

Guangxia Xu is currently a professor at Cyberspace Institute of Advanced Technology of Guangzhou University in Guangdong, China. She received the M.S. and Ph.D. degrees in computer science from the Chongqing University, Chongqing, China in 2006 and 2011, respectively. Her research interests include information security and network management, Big Data analytics for network security, and Blockchain technology. She is a committee member at the Blockchain of CCF, IEEE Senior Member and ACM member.



Zhihong Tian is currently a Professor, and Dean, with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong Province, China. He received his B.S., M.S., and Ph.D. degree in Computer Science and Technology from Harbin Institute of Technology, Harbin, China in 2001, 2003, and 2006 respectively. He is honored as Pearl River Scholar in Guangdong Province. He is also a part-time Professor at Carlton University, Ottawa, Canada. Previously, he served in different academic and administrative positions at the Harbin Institute of Technology. He has authored over 200 journal and conference papers. His research interests include computer networks and cyberspace security. His research has been supported in part by the National Natural Science Foundation of China, National Key research and Development Plan of China, National High-tech R&D Program of China (863 Program). He also served as a member, Chair, and General Chair of a number of international conferences. He is a Distinguished Member of the China Computer Federation, and Senior Member of IEEE.

Sergey Gataullin graduated from the State University of Management. He is an employee of the Financial University, Government of Russia, and a candidate of economic sciences. He is a specialist in the field of mathematical methods of decision-making, economic, and mathematical modeling. Since 2016, he has been engaged in research and administrative work at the Faculty of Information Technology and Big Data Analysis of the Financial University.