CCBRSN: A System with High Embedding Capacity for Covert Communication in Bitcoin

Weizheng Wang, Chunhua Su

CCBRSN: A System with High Embedding Capacity for Covert Communication in Bitcoin

Weizheng Wang[1] and Chunhua Su[1(✉)]

Division of Computer Science, University of Aizu, Aizuwakamatsu, Japan
chsu@u-aizu.ac.jp

Abstract. Covert communication has been using to prevent confidential information from being leaked to an unintended receiver. In this paper, we present a general purpose novel methodology for blockchain-based covert communication system design to be used in Bitcoin environment. Blockchain is a distributed system which combines P2P network, consensus protocol, encryption algorithm to complete the first reliable cryptocurrency system Bitcoin. According to the high security and convenient access of this technology, many applications based on Blockchain such as smart contracts, distributed cloud storage have been developed. However, in the field of covert communication, there are few researches are applied in Blockchain. Therefore in this paper, we propose a system called Covert Communication based on Bitcoin Regtest Self-built Network(CCBRSN), which takes Blockchain as a covert communication channel and embeds encrypted messages into Blockchains addresses to transmit. In this model, users can transmit covert messages via Blockchain mutually and fast. Finally, we provide experimental analysis for our proposal to show that it is suitable for practical application.

Keywords: Blockchain · Bitcoin · Cryptography · Covert Communication.

1 Introduction

Covert communication targets at hiding wireless transmissions, which meets the ever-increasing desire of strong security and privacy. In a typical covert communication system, a transmitter (Alice) intends to communicate with a legitimate receiver (Bob) without being detected with a warden (Willie), who is observing this communication [1]. Hence a reliable communication channel is the most important thing for covert communication, which can level up the security of critical communication or be confidential during the transmitting process.

Covert communication has two important parts—Cryptography and Steganography. Cryptography encrypts the plaintext to ciphertext, which can protect users' privacy. The method of steganography conceals the existence of the message, to make them unintelligible. Simultaneously, channel is crucial for steganography. In practical use, many channels have been tried as covert channels for steganography, for example, Covert Channels in the HTTP Network Protocol

[2], Covert Channels in IPv6 [3], A novel covert channel based on length of messages [4] and so on. Whereas, most of them are not stable, such as HTTP, if we meet an error during communication, maybe the transferring message is lost forever. Meanwhile, with the advance of decryption technology, the security of these channels can not be ensured in the future. All in all, it is still hard for us to find a high-quality and securable medium for steganography.

On the other hand, Blockchain is becoming more and more popular. As a highly decentralized, open and transparent distributed database structure, Blockchain was first introduced in the paper [5] which was published by Nakamoto in 2008. Blocks which are in a sequential order consist of the Blockchain. Every list of the transaction which is traded and packed is all recorded in a block. To maintain the runtime of Blockchain, every main node in the Blockchain real-time updates a separate global ledger. At the same time, all the main nodes are also competitors, they use their computing power to calculate a puzzle. In this race, the faster one will be the winner and get the authority of bookkeeping. If a winner in a mining race tries to tamper the data of one block or one transaction to get more profit, then the hash value of this block and the subsequent blocks will be changed immediately. After receiving this new-generated block, the honest nodes will compare the new ledger with owns and perceive the falsity of the chain. At last, they will refuse this published ledge and rollback. Based on security and openness, many applications choose to build up on the Blockchain, for example, smart contracts in the Ethereum [6], intelligent transport system [7], e-voting system [8]. In consideration of the above advantages of Blockchain, we also take Blockchain as a tentative covert communication channel.

There are plenty types of Blockchains in the world. Why do we select the Bitcoin as our experiment communication medium? There are the following four reasons: (1) huge computing power in the Bitcoin's network ensures the safety of Bitcoin, and until now the hash rate has already reached 92,468,911(TH/s) [9]. If an attacker attempts to take control of Bitcoin's network and do some bad things, it means he must get the 51% resource of the computing power. However, a regular GPU of PC can just provide 50—60(MH/s). Therefore, as a communication channel, Bitcoin is relatively reliable. (2) no matter where you are and what you have, as long as you can get access to the Internet, then you can use tens of thousands of applications that serve as light nodes in the Bitcoin to conduct transactions. Compared with other Blockchains, Bitcoin is more convenient as a channel. (3) cheap—you can deliver a million dollars' transaction with only a few dollars' transaction fees. (4) everyone can join or leave the network as they will, and this is an anonymous mode, you needn't concern about the disclosure of your privacy[10].

1.1 Related Works

Covert communication can be traced back to the steganography which was proposed in the 16th century's book—"Steganographia". With the development of the Internet, covert communication began to spring up in the late 20th century and now is widely used in the field of digital communication and network security.

Covert communication can be generally divided into two sides—Steganography and digital watermarking.

In terms of Steganography, there are many mature technologies, such as in the paper [11] suggested a way that uses a class of new distortion functions known as uniform embedding distortion function (UED) for both side-informed and non side-informed secure JPEG steganography, and in paper [12] also proposed an enhanced least significant bit modification technique for audio steganography, finally paper [13] proposed method creates an index for the secret information and the index is placed in a frame of the video itself.

On the other side, the digital watermarking means capable of carrying such information as authentication or authorisation codes or a legend essential for image interpretation, which is also an efficient way for covert communication. In 1997, Ingemar J. Cox et al. proposed a NEC algorithm [14] that combines the author's identification code with the image hash value to generate a key as a seed generation sequence, and then DCT transforms the image. This algorithm not only reduces the redundancy of the video signal, but also guarantees the robustness and security of the algorithm. Bender W suggested a digital watermark based on statistics [15], which selects a certain number of arbitrary pairs of image points in the image. When the brightness of one point in each pair of image points increases, the brightness of the corresponding other point decreases. To achieve the loading of the watermark, this method has good concealment and strong resistance but is not suitable for images with only a small amount of arbitrary texture.

1.2 Motivation

Although there are many methods for covert communication, however, we can only find a few experiments and applications on the Blockchain. In paper [16], Juha Partala et al. firstly suggested a method of submitting covert messages through a Blockchain considered as a payment platform. The overview of his ideal is to separate the message into unit bit and use the LSB(Least Significant Bit) of the address to match the unit bit, if the unit bit accords with the LSB then we use this address to do a transaction, repeat previous steps after this transaction is recorded into a block, finally the receiver can restore the message in a specific order.

This is an innovative way for covert communication in Blockchain, but it still exists some problems as follows:

1. In paper[16], they only provide a simplified Blockchain model hypothetically and discuss their scheme all in theory, in practical use it is hard to verify the feasibility of their named BLOCCE scheme.

2. A block is produced in Blockchain will cost users several minutes, if you want to transmit all bits of the message but one block only has one transaction, the whole process will need a long time to be finished. The timeliness of the message can't be promised.

3. Every transaction needs transaction fee, if the number of transaction is huge, you should pay a colossal sum of money. The Performance-to-Price ratio

of this communication may be low. Although there are some Blockchains which don't need transaction fee, it seems this problem could be solved. In fact, most of them are not the mainstream, we can't ensure their security.

Hence, we try to find a more practical and efficient way for covert communication on Blockchain.

1.3 Our Contributions

In this paper, we propose a scheme—CCBRSN which realizes covert communication on the Bitcoin's Regtest network. We also base on this scheme to develop a visualized operating system for the users. The users who join this network and use our devised tools can exchange messages safely, efficiently, conveniently. The main procedure of our ideal is to use DES and Base58 to encrypt and code the message successively, then we embed the ciphertext into a set of addresses in order, finally, we employ this set as output to conduct a transaction. After a transaction, a file for decryption will be produced. When the transaction is recorded into the Bitcoin ledger, the opponent can import the above-mentioned file into our tool to track this particular transaction and decrypt the ciphertext automatically. There are two remarking characteristics of our measures. One is we increase the embedded rate of address, which means one address can be embedded more information. The other feature is a message needn't be separated into multiple transactions, we can only use one transaction to transmit the whole address set(message) in most of the time. Although this system is proposed for Bitcoin, while other Blockchain only if which adopts address can also use this model to construct a covert communication system.

1.4 The Organization of Remaining Paper

The rest of the paper is organized as follows. Section 2 explains the preliminaries for the rest of the paper, such as the structure of Blockchain we use, the configuration of Regtest Network, the method of encryption, coding and threat model. Section 3 describes the detail of our proposed CCBRSN Scheme. The experiment environment and evaluation are presented in section 4. Finally, we conclude and propose future work in section 5.

2 Preliminaries

2.1 The Structure of Blockchain

The birth of Bitcoin creates the conception of Blockchain. All the nodes in the Blockchain maintain a public ledger. When each transaction is produced, it will be verified by all nodes to ensure no error. After checking, this transaction will be packed into a block in the structure of the Merkle tree and added into a decentralized ledger. The relation between transaction and block is illustrated in Figure 1.

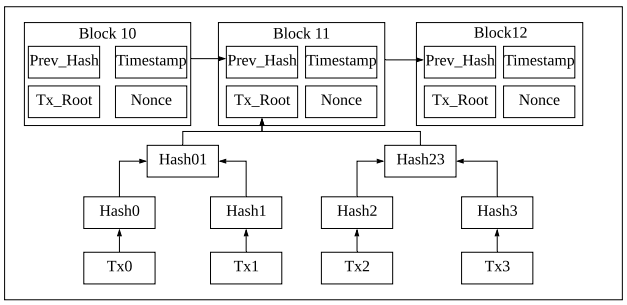

Fig. 1. The structure of Block.

The biggest difference between Bitcoin and other systems is whether or not exists a central trusted authority. Blockchain is a decentralized system, in this network trust is achieved by every participated node. If there is an attacker who attempts to take charge of the system, firstly he must occupy the 51% computing power of the entire Blockchain. However, it is difficult for common people to get enormous computing power. For this reason, Blockchain is quite safe, people can feel free to do the transaction on them.

Blockchain technology has attracted enormous investors and researchers from fields of healthcare, finance, transportation, government and so on. Until now, there are various applications that are built on the Blockchain, such as smart contracts, intelligent transportation, and identity verifying, etc. Bitcoin is a first-generation Blockchain, Ethereum broke the mold by becoming the first-ever second-generation Blockchain. Ethereum revolutionized the crypto-space by bringing in smart contracts on the Blockchain. Smart contracts were first conceptualized by Nick Szabo. The idea is simple, have a set of self-executing instructions between two parties that dont need to be supervised or enforced by a third-party. The idea seems pretty straightforward. However, smart contracts enabled Ethereum to create an environment wherein developers from around the world could create their decentralized application aka Dapps [17].

2.2 The Configuration of Regtest Network

Bitcoin and most other cryptocurrencies have 3 modes of operation. Mainnet is the network which is used as the official version, and it has value. All real transactions happen on this network, people get paid or pay using Mainnet.

Testnet, a network which has almost the same rules (some opcodes are forbidden on Mainnet, while this restriction is lifted on Testnet) as Mainnet. It has peer discovery, that is it can find peers on the Testnet network, similar to Mainnet, and a peer-to-peer (P2P) network is running it.

Regtest is a private Blockchain which has the same rules and address format as Testnet, but there is no global p2p network to connect to [18]. It is convenient because this feature makes us easily to build Bitcoin network in local.

After dowloading the Bitcoin core client, only one thing we need do configuring Bitcoin.conf file to change network from Mainnet to local Regtest network.

```
E:\bitcoin-0.17.1\bin>bitcoind
2019-11-01T15:51:06Z Bitcoin Core version v0.17.1 (release build)
2019-11-01T15:51:06Z InitParameterInteraction: parameter interaction: -whitelistforcerelay=1 -> setting
-whitelistrelay=1
2019-11-01T15:51:06Z Warning: Config setting for -rpcbind only applied on regtest network when in
[regtest] section.
2019-11-01T15:51:06Z Validating signatures for all blocks.
2019-11-01T15:51:06Z Setting
nMinimumChainWork=0000000000000000000000000000000000000000000000000000000000000000
2019-11-01T15:51:06Z Using the 'sse4(1way),sse41(4way),avx2(8way)' SHA256 implementation
2019-11-01T15:51:06Z Using RdRand as an additional entropy source
2019-11-01T15:51:06Z Default data directory C:\Users\lifewwz\AppData\Roaming\Bitcoin
2019-11-01T15:51:06Z Using data directory C:\Users\lifewwz\AppData\Roaming\Bitcoin\regtest
2019-11-01T15:51:06Z Using config file C:\Users\lifewwz\AppData\Roaming\Bitcoin\bitcoin.conf
2019-11-01T15:51:06Z Using at most 125 automatic connections (2048 file descriptors available)
2019-11-01T15:51:06Z Using 16 MiB out of 32/2 requested for signature cache, able to store 524288
elements
2019-11-01T15:51:06Z Using 16 MiB out of 32/2 requested for script execution cache, able to store
524288 elements
2019-11-01T15:51:06Z Using 8 threads for script verification
2019-11-01T15:51:06Z scheduler thread start
2019-11-01T15:51:06Z libevent: getaddrinfo: nodename nor servname provided, or not known
2019-11-01T15:51:06Z Binding RPC on address :: port 18443 failed.
2019-11-01T15:51:06Z HTTP: creating work queue of depth 16
```

Fig. 2. The successful execution of Regtest network.

From the Figure 2, we can see the program shows Bitcoin client version and the information of network creation. It means the local network have been established successfully. Then we can use bitcoin-cli commands to execute some operations in the Bitcoin network, such as send transactions or get block information.

2.3 The Method of Encryption and Coding

In this model, we do two operations on original message, one is DES encryption, the other one is Base58 encoding.

In the covert communication, it is necessary for us to encrypt the message firstly in case of disclosure. At present, maybe the encryption of DES are not strong enough very safe, an attacker who attempts to use brute-force can crack the ciphertext. As a result, many corporations and individuals tend to choose other encryption such as AES, RSA. In the consideration of achieving a balance between simplicity and security, the DES encryption is better. Because the outcome of DES encryption is shorter than AES or 3DES, and the security of message can be ensured in some ways. The length of ciphertext is of great importance for transaction. In theory, the shorter one needs less addresses to match. Hence, we select DES as our encryption scheme, which can increase the huge efficiency of embedded rate.

After encrypting, then we should use Base58 to encode our encrypted message. In Bitcoin, the adaptation of coding scheme is Base58, if we want to embed

ciphertext into addresses, the absolutely necessary thing is to unit the coded system. Then we can make some matches.

Finally, the system will produce a decrypt file for receivers to find the original message. We put another DES encryption to this decrypt file again, which can resist the cryptoanalysis from the leakage of the this file.

2.4 Threat Model

Here, we assume there are some attackers in our model who attempt to tamper or intercept message, cryptoanalyse transaction.

If attackers attempt to tamper data on the Bitcoin, it means they should create enough fake identities, which can repel real nodes on the network by a majority of votes. Then fake nodes can reject the receive or transfer blocks, effectively preventing other users from accessing the network. In the relatively large-scale Sybil attack, the premise is that when the attackers have controlled most of the computer network or hash rate, they can carry out the system attack covering 51%. In this case, they can easily change the order of the trades and prevent the trades from being confirmed. They can even take over and reverse transactions, leading to a double payout problem. But Bitcoin mining is intense and highly rewarding, most of the miners are keen on legitimate mining methods instead of trying to conduct Sybil attack. Bitcoin as a covert communication medium, which can guarantees the truth of the data[19].

All the transactions in the Blockchain are transparent, the attackers also can witness the generation of these transactions. If a user continues to transmit overlong length of messages, the attackers may notice the continuous transactions with multiple output addresses. Maybe they will analyse these transactions, even if they distinguish some transactions from the same sender, they still can't understand how to restore the addresses to the ciphertext. Because our model CCBRSN takes a special rules to embed messages into addresses and use DES to encrypt the decrypt file again. The only way to crack the ciphertext is to intercept the decrypt file the model produced and get access to our tool at the same time. In fact, it seems very impossible unless the user reveals initiatively.

3 The proposed CCBRSN Scheme

In this section, we give an exhaustive description of our scheme called CCBRSN. Firstly we introduce the configuration of the Regtest network in Bitcoin. Then we describe the overview of our method. At last, we will talk about the process of embedding and restore in detail.

3.1 General Overview of CCBRSN

In our scheme, User A wants to send an encrypted message to User B through the Bitcoin, however, User A doesn't expect this message is revealed or decrypted to anyone other than User B. Due to consensus algorithm in Bitcoin, if User A

completes all the process of sending, anyone even User A himself can't change the content of the message. For malicious guys, they also have no ability to tamper the text of the message because of this feature in Bitcoin. The general procedure of our scheme is as follows:

1. User A inputs a message m that he wants to send.

2. User A enters a password k for DES encryption, the outcome of encrypting m is $DES_k(m)$. Then $DES_k(m)$ is encoded into $Base58(b)$ by $Base58$.

3. User A applies $ECDSA$ to generate a pair of private and public key$(s_k^{(1)}, p_k^{(1)})$

4. User A starts with the $p_k^{(1)}$, computes the SHA256 hash and then computes the RIPEMD160 hash of the result, finally uses $Base58$ to produce an address $a^{(1)}$.

5. User A compares each bit of $Base58(b)$ with each bit of $a^{(1)}$. If the match is successful, then User A will record the corresponding indexes of $Base58(b)$ and $a^{(1)}$ $set^m[[1, 2, ..., n], ..., [1, 2, ..., n]]$ and $set^a[[1, 2, ..., n], ..., [1, 2, ..., n]]$.

6. User A replaces the corresponded bits in $Base58(b)$ with *, then $Base58(b)$ will be transformed into $Base58(b1)$. User A continues to use $Base58(b1)$ to repeat the above steps until all bits in $Base58(b1)$ are matched.

7. User A submits a transaction whose output addresses are $a^{(1)}, a^{(2)}, a^{(3)}, ..., a^{(n)}$. Transaction fee of each address are in a address generated time order. After sending process, a file $File$ will be produced.

8. For the protection of information in the file, we should use DES encryption with a defined key in advance to encrypt the $File$ to $encFile$ again.

9. Until this transaction is packed into a block, User B can import file $encFile$ to restore the message.

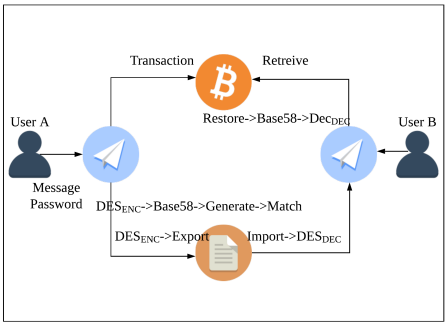

Fig. 3. The process of execution.

During the entire transaction, User A only pays a little money for transaction fee. Because all pairs of private and public key are all in User A's charge, he can transfer back the money. Although we base this system on Bitcoin, other Blockchains who accept address can also use this model to build a new covert

communication system on their own. The detailed steps have been illustrated in the Figure 3.

3.2 Messages Embedment

In this section, we explain the detailed steps of the embedding algorithm. Firstly, we assume that User A can transmit encrypt file to User B in a public channel. Although somebody could intercept this file, however, it is hardly possible for him to find our tool for decryption. In the initial stage, we use DES to encode the original messages. Now DES is not safe which can be cracked by some analysis. Considering the length of the outcome, we still choose this encryption algorithm. Like $Base64$, $Base58$ is also a specific coding scheme that is employed in the Bitcoin system. To make our ciphertext correspond with address, we take the same way to encode text.

In our model, we build our test experiment on the Bitcoin Regtest network. Even though, Regtest is developed in local and doesn't have all the abilities of Mainnet. It still can satisfy most of the requirements it needs, for example, the generation of public and private keys, generate address, transaction and so on.

The following algorithm1 is our embedding procedures.

Algorithm 1 Messages Embedding Algorithm

procedure EMBED(k, m)
  $c \leftarrow DES(k, m)$
  $b \leftarrow Base64(c)$
  $n \leftarrow 0$
  while b != $'**...**'$ **do**
    $n \leftarrow n + 1$
    $s_k^{(n)} \leftarrow Random(2^0, 2^{256})$
    $p_k^{(n)} \leftarrow KeyGen(s_k^{(n)})$
    $a_k^{(n)} \leftarrow BASE58(RIPEMD160(SHA256(p_k^{(n)})))$
    $Flag \leftarrow False$
    for each bit_1 in $a_k^{(n)}$ **do**
      for each bit_2 in b **do**
        if $bit_1 == bit_2$ **then**
          $Flag \leftarrow True$
          $setAddr_{index} \leftarrow IndexOf(bit_1)$
          $setMsg_{index} \leftarrow IndexOf(bit_2)$
        end if
      end for
    end for
    if $Flag == True$ **then**
      $setAddr_{addr} \leftarrow a_k^{(n)}$
    end if
  end while
  if $setAddr_{addr}$!= $NULL$ **then**

$TxID \leftarrow$ Transaction($setAddr_{addr}$)
$encFile \leftarrow$ Write($Enc('wwz12345')$,
$setAddr_{index}, setMsg_{index}, TxID$)
end if
end procedure

3.3 Messages Extraction

Extraction is the reverse process of embedment. User B will import the received file into the system to restore some important parameters. Then User B uses the value of $TxID$ to locate the corresponding transaction information and find all the addresses. Finally, User B can decode and decrypt the ciphertext to plaintext. The detailed procedure of extraction is described in Algorithm2.

Algorithm 2 Messages Extraction Algorithm

procedure EXTRACT($File$)
$File \leftarrow Dec('wwz12345', encFile)$
$TxId \leftarrow Restore(File)$
$setAddr_{index} \leftarrow Restore(File)$
$setMsg_{index} \leftarrow Restore(File)$
$k \leftarrow Restore(File)$
$setAdd_{addr} \leftarrow FindTx(TxID)$
$b \leftarrow Match(setAddr_{index}, setMsg_{index},$
$setAddr_{addr})$
$c \leftarrow deBase64(b)$
$m \leftarrow Dec(c, k)$
end procedure

4 Experiment and Evaluation

4.1 Experiment Environment

Our model is deployed in Windows 10. The main used tools are Bitcoin Core 0.18.1 client and python 3.6. Based on the scheme we proposed, we use python and the package of bitcoinlib to develop a covert communication system which connects to the Bitcoin Regtest Network. Users can use this system to transmit messages in this self-built network at any time. For the evaluation of our model, we choose 10 different length of text from 5 bits to 50 bits. Then inputting random generated data to the tool we developed, every type of text is tested 100 times for the accuracy of the experiment. Then we calculate the average value for our experiment.

4.2 Evaluation Results

We are from three aspects to build experiments for the evaluation of our model.

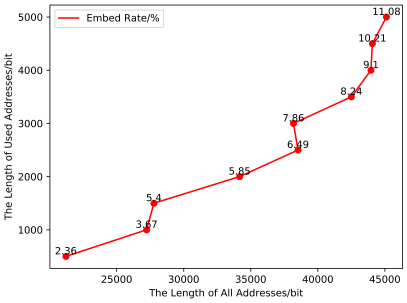

Fig. 4. The Embed Rate in CCBRSN

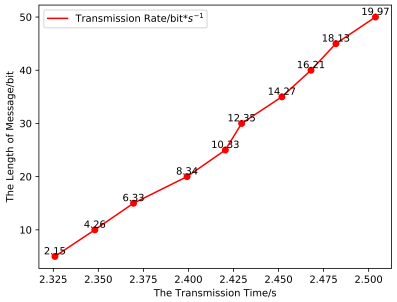

Fig. 5. The Encrypt Rate in CCBRSN

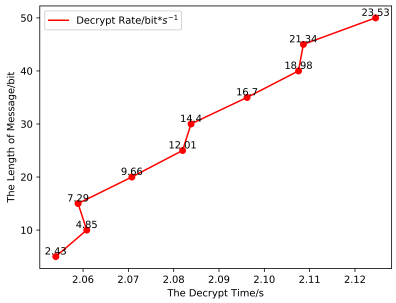

Fig. 6. The Decrypt Rate in CCBRSN

1.Embeded Rate: Firstly, we use our dataset to analyse the embed rate of generated addresses. As we can see from Figure 4, although there is a fluctuation during the whole process, overall the reuse rate of address becomes higher with the increment of text length. The increment is very apparent, at last, the embed rate even reaches to 11.08%.

2.Transmission Rate: We record all the time from inputting data to the generation of decrypt file. Figure 5 describes the transmission rate between the length of messages and the transmission times. At the beginning, the encrypt rate is a little low which is just around 2.15 bit/s. Because there are 34 bits text in address, whereas, some bits in an address are also repetition. If we attempt to pair address and text, the short length of sending text need the same number of addresses like long ones. Hence the first step will cost fixed time, then the processing time will slow down and have a little difference. Once the length of text increases a lot, the rate of address utilization will be raised, then the embed rate will have a huge gap. As we can see from the figure, the fastest transmission rate can be up to 19.97 bit/s. Therefore, at first, the transmission rate is very low, but as length increases, the transmission rate is getting faster. In conclusion, the

increment of bits corresponds with the trend of the time, and the transmission time is in the tolerance interval.

3.Decryption Rate: Figure 6 shows the decrypt rate in CCBRSN. We can see decrypt rate is the same as encrypt rate, which is very low at the beginning, then high at the ending. The reason for explaining this phenomenon is the same as embed rate. However, in this stage, the difference is restoring message dominates the main time, other things only have a small effect on it. Hence, the tiny increment of text size leads to enormous improvement in decrypt rate, which even achieves 23.56 bit/s. Although the decrypt rate is not fast at the beginning, but there are always some delays in communication, decrypt time is around 2s which is also considered as acceptable.

4.3 Compared with other schemes

Few related papers also take blockchain as a covert channel medium. In the paper[16], they proposed a pioneering concept attempts to conduct covert communication in Blockchain and give us a simple example.BlBasuki et al.[20] suggested joint steganography by utilizing blockchain-based transaction steganography and image steganography to achieve a secure and secret communication medium. we try to use the following tables to analyze the pros and cons of these methods

Method	Capacity/transaction	Security	Blockchain networks
BLOCCE	2000 bits	+	Any
JTISHCC	29 bits	++	Ethereum
CCBRSN	68000 bits	++	Any(default: Bitcoin)

Table 1. TRANSACTION STEGANOGRAPHY COMPARISON

As we can see from Table 1, in the aspect of capacity, our scheme-CCBRSN has the overwhelming advantages of the other methods. Due to the reuse features of the addresses, if possible CCBRSN can make use of every bit in an address(most of the addresses have 34 bits). The max number of addresses in a transaction is usually between 2000 and 3000. When it refers to security, our scheme uses Des encryption and special coding rules to ensure the privacy of our content. Simultaneously, only if the blockchain who has addresses for the transaction, our schemes can be applied to, therefore the appliance is rarely widespread.Based on the above analysis, CCBRSN is suitable for covert communication.

5 Conclusions and Future Work

In this paper, we present a method which is called CCBRSN. This method firstly encrypts and encode plaintext into a specific type, then embed this type of mes-

sage into Bitcoin's addresses, finally construct a transaction that includes these addresses to transmit information. If this transaction is finished successfully, a decrypt file will be produced. If a user who owns the decrypt file can input it into the same system, then the plaintext will be restored.

Although our way has successfully implement covert communication on Bitcoin, It still needs further improvement. For example: 1) Our model is just based on local network, so we don't take deeply consideration of the crowd of transaction, the latency of the block generation and so on which really exists in public network. 2) We use multiple match to greatly improve the efficiency of transmission. However if a user attempts to send a huge size of message such as document, the number of address may reach thousand or even ten thousand, then the size of a transaction will exceed the rated value. As a result, this transaction can't be generated. 3) Decrypt file is significant for receivers, users will use this specific file to restore message. Nonetheless, we don't know people how to transmit it, maybe in a public communication way. If an attacker intercepts this file and just right has access to this tool, the security of transmission can't be ensured. 4)In the future research, we can attempt to apply our model to other Blockchain systems.

For the improvement of our scheme, we will attempt to take some countermeasures to solve the above mentioned problems in our future research.

Acknowledgement

This work is partly supported by JSPS Kiban(B) 18H03240 and JSPS Kiban(C) 18K11298.

References

1. Weile Zhang, Nan Zhao, Shun Zhang, and F Richard Yu. Multi-antenna covert communications with random access protocol. *arXiv preprint arXiv:1907.07481*, 2019.
2. Erik Brown, Bo Yuan, Daryl Johnson, and Peter Lutz. Covert channels in the HTTP network protocol: Channel characterization and detecting man-in-the-middle attacks. In *5th European Conference on Information Management and Evaluation, ECIME 2011*, 2011.
3. Norka B. Lucena, Grzegorz Lewandowski, and Steve J. Chapin. Covert channels in IPv6. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006.
4. Liping Ji, Wenhao Jiang, Benyang Dai, and Xiamu Niu. A novel covert channel based on length of messages. In *Proceedings - 2009 International Symposium on Information Engineering and Electronic Commerce, IEEC 2009*, 2009.
5. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System — Satoshi Nakamoto Institute. Technical report, bitcoin.org, 2008.
6. Gavin Wood. Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014.

7. Yong Yuan and Fei Yue Wang. Towards blockchain-based intelligent transportation systems. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2016.
8. Michał Pawlak, Aneta Poniszewska-Marańda, and Natalia Kryvinska. Towards the intelligent agents for blockchain e-voting system. *Procedia Computer Science*, 141:239–246, 2018.
9. The number of tera hashes per second in the bitcoin network. `https://www.blockchain.com/zh-cn/charts/hash-rate`. Accessed October 17, 2019.
10. Qiheng Zhou, Huawei Huang, Zibin Zheng, and Jing Bian. Solutions to scalability of blockchain: A survey. *IEEE Access*, pages 1–10, 2020.
11. Linjie Guo, Jiangqun Ni, and Yun Qing Shi. Uniform embedding for efficient jpeg steganography. *IEEE transactions on Information Forensics and Security*, 9(5):814–825, 2014.
12. Muhammad Asad, Junaid Gilani, and Adnan Khalid. An enhanced least significant bit modification technique for audio steganography. In *International Conference on Computer Networks and Information Technology*, pages 143–147. IEEE, 2011.
13. R Balaji and Garewal Naveen. Secure data transmission using video steganography. In *2011 IEEE International Conference on Electro/Information Technology*, pages 1–5. IEEE, 2011.
14. Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.
15. Walter Bender, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. Techniques for data hiding. *IBM systems journal*, 35(3.4):313–336, 1996.
16. Juha Partala. Provably secure covert communication on blockchain. *Cryptography*, 2(3):18, 2018.
17. Different blockchains: Ethereum vs cosmos vs hyperledger and more! `https://blockgeeks.com/guides/different-blockchains/`. Accessed October 20, 2019.
18. How to set up a bitcoin regtest environment. `https://bisq.network/blog/how-to-set-up-bitcoin-regtest`. Accessed October 19, 2019.
19. Huakun Huang, Shuxue Ding, Lingjun Zhao, Huawei Huang, Liang Chen, Honghao Gao, and Syed Hassan Ahmed. Real-time fault-detection for iiot facilities using gbrbm-based dnn. *IEEE Internet of Things Journal*, 2019.
20. Akbari Indra Basuki and Didi Rosiyadi. Joint transaction-image steganography for high capacity covert communication. In *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 41–46. IEEE, 2019.