Secret and Trustable Communication Channel over Blockchain Public Ledger

1st Muhammad Fajar Sidiq
Department of Informatics
IT Telkom Purwokerto
Purwokerto, Indonesia
fajar@ittelkom-pwt.ac.id

2nd Fahrudin Mukti Wibowo
Department of Informatics
IT Telkom Purwokerto
Purwokerto, Indonesia
mega@ittelkom-pwt.ac.id

3rd Merlinda Wibowo
Department of Informatics
IT Telkom Purwokerto
Purwokerto, Indonesia
merlinda@ittelkom-pwt.ac.id

4th Akbari Indra Basuki
Research Center for Informatics
Indonesian Institute of Sciences
Bandung, Indonesia
akbari.indra.basuki@lipi.go.id

5th Iwan Setiawan
Research Center for Informatics
Indonesian Institute of Sciences
Bandung, Indonesia
iwan.setiawan@lipi.go.id

6th Didi Rosiyadi
Research Center for Informatics
Indonesian Institute of Sciences
Bandung, Indonesia
didi.rosiyadi@lipi.go.id

***Abstract*—Blockchain public ledger is a trusted medium that provides incorruptible data storage and traceable transactions. However, due to its public verifiability, it is hard to compromise two functionalities at once, a covert channel and a verifiable communication medium. The existing methods are either vulnerable to statistical analysis or lack trustable delivery notifications due to off-chain message delivery. This paper fills the gap by proposing a Multi-addresses Random Set Encoding (MaRSE) to increase message covertness by retaining the natural transactions pattern and resisting statistical attacks. The method uses on-chain message delivery to preserve blockchain-based verification that provides a trustable communication medium for the communicating parties. As a result, our proposed method is robust to a brute force attack due to its unpredictable transaction sequence and high number possibility of addresses combinations. The proposed system has been implemented on Ethereum rinkeby networks with the stegano-transactions being recorded within block numbers 4932833 to 4932893.**

***Index Terms*—Secret, Trustable, Communication, Blockchain, multi-addresses encoding, permutation.**

I. Introduction

A blockchain ledger is a reliable and trusted medium, established by three main components: public-key cryptography, distributed consensus algorithm, and immutable hash chain schema. As a result, blockchain provides immutable data storage and traceable transactions that verifiable by anyone using a cryptographic signature.

The most attractive use of blockchain is to preserve user privacy. In a blockchain, transaction validation is based on cryptography schema rather than user identity. Thus, everyone can issue a secret transaction while maintaining their privacy since the transaction only record the addresses of the sender and the recipient. The real identity of the person who makes the payment is unknown, so does with the recipients.

Blockchain privacy-preservation can be exploited to implement a covert communication channel that requires not only secret identity but also secret channel and content. Blockchain-based secret communication provides traceable communication which means it is possible to prove that the communication was happening and the recipient cannot deny that the sender has sent the message. As consequence, it can eliminate the distrust among the communicating parties.

Unfortunately, due to the low capacity of blockchain transactions to store the hidden message, some existing works use off-chain delivery to offer a larger capacity. Some of them use IPFS networks [1], images [2]–[4], or text files [5] for the off-chain delivery. The message is first encrypted while the encryption key is driven by the on-chain communication. Even though it increases the message capacity, the methods nullify blockchain's universal verifiability. As a result, the recipients might simply deny the receiving for any kind of reason such as due to network censorships, or corrupted files.

Meanwhile, other works that solely uses on-chain delivery suffers from unnatural transaction pattern or low message capacity [6]–[9]. The simplest way to increase message capacity is by excessively modify the transaction fields. The work in [6] uses transaction value to store up to 28.12 bits per transaction. Nevertheless, this technique gives unnatural patterns and prone to statistical analysis that reveal the hidden communication channel. In another hand, using only the destination address to store the hidden message will yield a low throughput. As being presented in [9], the method can only store single-bit data per destination address.

In this paper, we proposed Multi-addresses Random Set Encoding (MaRSE) to implement a covert communication medium

978-1-6654-3434-8/21/$31.00 ©2021 IEEE

over blockchain networks that satisfy both, the covertness of the secret channel and its trusted verifiability. The method encodes the hidden message into a series of blockchain transactions called stegano-transactions. The key point is to use multiple addresses at once, called a set address. It aims to improve message capacity and the secretness of the message by selecting a different subset for each session. Meanwhile, the use of on-chain message delivery ensures undeniable message delivery for the communicating entities.

The proposed method is robust against brute force and statistical analysis attacks due to several reasons as follows.

1) The address can be anything, from a blockchain account to a smart contract one. Consequently, it covers all types of addresses which is harder to analyze.
2) The method does not vary the transaction field value. Thus, it is hard to infer which transaction belongs to our covert channel by analyzing the irregularity. It can be a regular payment, one-time transaction, or even a smart contract transaction.
3) It is hard to infer the set addresses out of the existing blockchain addresses.
4) It has unpredictable sequences since any transactions can be inserted in between. As a consequence, there are unlimited combinations to infer the communication channel.
5) For each communication session, it is hard to infer the currently selected subset.

We further elaborate the MaRSE method as follow. Sections II presents the system design, while chapter III describes the robustness model. Chapter IV discusses the result and its evaluations. Finally, chapter V presents the conclusion.

II. System Design

A. System abstraction

For simplicity, we divide the system into three abstraction layers as depicted in Fig. 1, Message layer, Packet layer, and Transaction layer.

The message layer manages the message delivery into the blockchain ledger. In this layer, a new session key called a message key is generated to compute the new seed for subset selection. After sending a message, the sender computes a new message key for the next message by hashing the shared secret key with the block number (Bnumber) of the latest stegano-transactions (1).

$$message_{key} = hash(secret_{key}, Bnumber_{prev}) \quad (1)$$

$$seed_{key} = hash(Bnonce_{prev}, message_{key}) \quad (2)$$

In the packet layer, the message is breakdown into a per-character basis as a pair of source and destination addresses. The pair is chosen randomly from a subset address while the subset address is randomly selected from the main set address by using a random seed ($seed_{key}$). It is generated by hashing the message key with the previous block-nonce ($Bnonce$) where the stegano-transaction is being recorded (2). For the first character, the block nonce refers to the latest block of the previous session. The illustration regarding the selection of address-pair is depicted by Fig. 2.

For the last step, the transaction layer sends the stegano-transaction by using the selected pair of addresses into blockchain addresses.

B. Ethereum transactions model

In this paper, we limit our discussion to Ethereum blockchain. In Ethereum blockchain, every account consists of 20-bytes of hexadecimal characters or 160-bit binary values [10].

Given all of the possible Ethereum addresses as a set A. A simplified model of Ethereum transaction (t) can be expressed as $t = (A_S, A_D, G_L, G_P, V, I_D)$. Where $A_S, A_D \in A$, A_S represent the sender address and A_D represent the destination address. G_L, G_P, and V respectively represent the Gas limit, gas price, and transaction value. Meanwhile, I_D represents the extra input data of the transaction.

In a single address schema, a single user (u) might send several transactions so that the generated transactions will form a transaction set (T_u), $T_u = \{t_1, t_2, ..., t_t | t_i \in T_u\}$. Where i represent transaction counter of sender u. The user's address u_a acts as the sender address for every transactions in (T_u).

In multi-address schema, each user (u) has a set of addresses S_u, where $S_u = \{s_1, s_2, ...s_n | s_i \in A\}$. The transactions generated by the user u is the union of all of the transactions sent by each address in S_u such that $T_u = \{T_{u\_s1} \cup T_{u\_s2} \cup ... \cup T_{u\_sn}\}$.

C. Multi-addresses Random Set Encoding (MaRSE)

The MaRSE method uses address-based encoding to translate the hidden message into a pair of source and destination addresses. The set of source addresses refers to the sender's address S_u, while the set of destination addresses (D_u) is a set of destination addresses determined by the sender. The sender must share the destination address (D_u) along with the shared secret key to their intended recipient before sending any hidden message.

The address-based encoding is a mapping function of $y = f(x, seed_{key})$ that maps a full set of n-bit of binary number into a set of address S consisting of 2^n elements. For example, a set of 2-bit binary number x, $x = \{00, 01, 10, 11\}$, can be encoded into a set of address $S_A = \{s_1, s_2, s_3, s_4\}$.

For harder predictability, instead of using the entire set address to encode the hidden message, the MaRSE method only picks some of the addresses or the proper subset (S_S and D_S) to encode the message on a per-character basis. Where S_S is the proper subset of source addresses ($S_S \subset S_u$) and D_S is the proper subset of the destination addresses ($D_S \subset D_u$). For each delivered transaction, the method always recomputes the

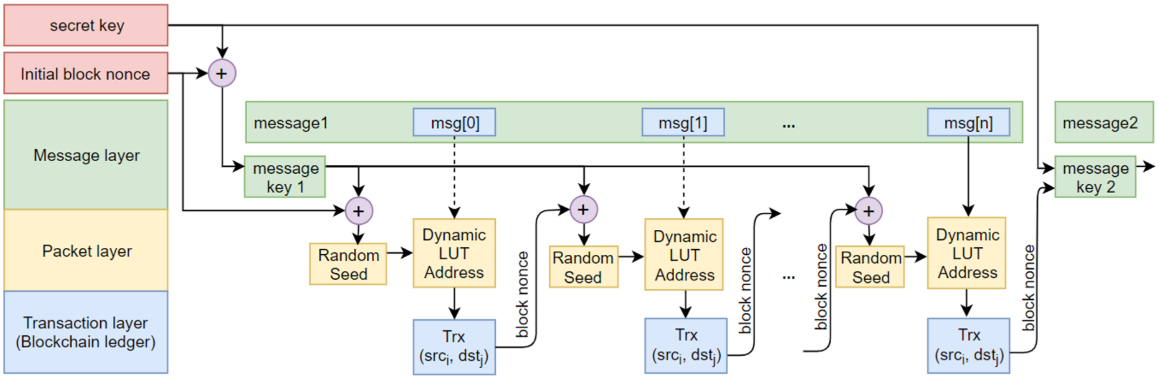


Fig. 1. MaRSE Abstraction layer

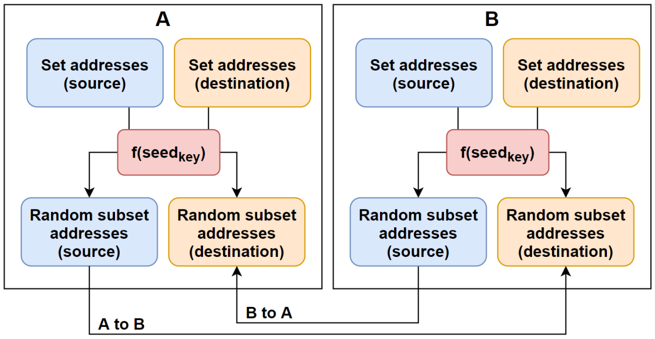

A
B
Set addresses (source)
Set addresses (destination)
f(seed$_{key}$)
Random subset addresses (source)
Random subset addresses (destination)
A to B
B to A

Fig. 2. Scanning for hidden message

encoding lookup table (LUT) to provide high randomness and increase the difficulty against brute-force attacks.

D. Valid Stegano-transactions

A valid stegano-transaction (Vt) is a transaction sent by a source addresses s_i to a destination address d_j, where $s_i \in S_S$, $d_j \in D_S$. Each time the sender sends a stegano-transaction, the random set encoding changes the encoding function (y) by using a new random seed such that $y = f(x, seed_{key})$, to generate a new pair of a proper subset S_S and D_S.

Random transactions can be inserted in between two valid stegano-transactions by carefully selecting source addresses and destination addresses (s_i, d_j). If the sender address is chosen from the selected set $(s_i \in S_S)$ then the destination address cannot be one of the destination set $(d_j \notin D)$. If the sender address is not within the selected set $(s_i \notin S_S, s_i \in S_u)$ then the destination address must be the member of destination set (D_S). Herein, the receiving parties can filter out the filling transactions from the valid stegano-transactions.

For the delivery, we use the filling ratio (f_r) to ensure a non-consecutive delivery. For example, if we choose $f_r = \{0.5\}$, then a valid stegano transaction will be sent after sending a filler transaction. For a message consist of 64 characters, the hidden message will be delivered within ≈ 128 transactions.

III. System Robustness

The robustness of the proposed method refers to its resistance against brute-force attacks that aim to reveal the hidden message. We divide the security model based on the abstraction layers. In the packet layer, we show that the MaRSE method is hard to decipher since it follows a set permutation schema with a high degree of variations and no inter-session correlation. In the message layer, it is hard to find the correct message sequences because the length of the message and the starting point of every message is unknown.

A. Packet layer security

The packet layer consist of two difficulties: finding the set addresses from all of the existing Ethereum addresses, and finding the proper subset addresses for a given communication session.

1) Finding the set addresses: Directly brute-forcing set addresses is a daunting task due to two reasons, the possible number of variations is too high and there is no inter-session correlation. Given a blockchain ledger L that record a sequence of blocks B, $(L = \{b_1, b_2, ...b_n\} | b_i \in B)$, and each block b_i consist of n-transactions from source addresses s_j to destination address d_k such that $b_i = \{t_1(s_1, d_1), t_2(s_2, d_2), ..., t_n(s_j, d_k) | t_i \in T\}$, proving that a set of source addresses are belong to the same sender is hard.

Considering the sender recomputes a new subset for each session, finding the correct subset cannot be solved in polynomial time since there is no inter-session correlation. The same address might be part of the subset for one session, but not for the next session. The same also applies to the destination addresses.

2) Finding the selected subset for each session: The easier brute force attack can be done on a per-session basis. For set address consist of m-elements, theoretically, there will be $C(2^{160}, m)$ possible combinations to guess. However, given an observation period (P), the number of unique source and destination addresses are smaller than all possible addresses (A). The actual number of unique source and destination addresses within a certain period (P) can be denoted as Q_S and Q_D. Thus, the total combinations between the multiple sources address (m_S) and multiple destinations address (m_D) can be rewritten into (3).

$$C_{Tot} = \sum_{i=1}^{log_2 m_S} C(Q_S, 2^i) * \sum_{i=1}^{log_2 m_D} C(Q_D, 2^i) \tag{3}$$

The m_S, m_D are determined by the sender, while Q_S and Q_D always vary along the time according to the activity within the blockchain networks.

Considering that the attackers do not have information regarding the size of the subset addresses, they must first guess the size of the subset before brute-forcing the message. Thus the total possible combination to guess the message will be sum up from 1 to $log_2 m_S$ and $log_2 m_D$ for respectively the source addresses and destination addresses

B. Message layer security

We classify the security of the message layer into two cases: 1) if the set address of the sender is unknown, and 2) if the set is known. Ideally, the second case will never occur, unless, it was intendedly leaked by one of the communicating parties. We do not pursue this direction since it is considered a human error instead of a systemic one.

In general, the message layer security inherits the security of the packet layer. Given a hidden message H with n-packet length, the adversaries must try the combination of $n * C_{Tot}$ to brute force the address combinations and reveal the true message H. However, it is valid only for a consecutive delivery.

In a non-consecutive delivery, for a period of observation blocks (O_B) that contains v-transactions, finding the correct t-number of stegano-transactions (Vt) yields $C(v, t)$ possible combinations. It amplifies the difficulty to find the hidden message into $C(v, t) * C_{Tot}$.

By using a non-consecutive delivery, it becomes harder to find the hidden message. The number of transaction and the number of stegano transaction per message under a particular observation period (O_B) have no exact boundary. The value of v and t can be any number n, $n \in \mathbb{N}$. The sender has the freedom of how to send the packets, whether within a consecutive block or within a several weeks or months interval. Consequently, the value of v ranging from v_{min} to v_{max}, where ($1 < v_{min} < v_{max}$), ($v_{min} < v_{max} < nB$), and nB is the latest block number of the blockchain ledger. However, in a real case the value of $V_{max} << nB$.

The value t cannot be exactly determined because the message length always varies from one message to another. The RSE method uses the message length to differentiate between messages. It encodes message length with the same space capacity as the message content. As a result, it is hard to tell whether a particular packet contains the message content or the message length. The value t can ranges from as small as 1 packet to $2n$ packets. Considering that n refers to the capacity of one address (Section II-C) then $2n$ refers to a pair of source-destination addresses. The $2n$ value also represents the space capacity of message content or message capacity per transaction.

C. MaRSE security

The total difficulties to brute-force a hidden message if the set address of the sender is unknown $(MaRSE_{diff.})$ can be expressed by (4). The C_{Tot} refers to packet layer security while the $C\left(v_i, t_j\right)$ refers to non-consecutive delivery security. The $2n$ refers to the pairs addresses and v_{max} refers to the maximum transaction number within the observation period (O_B).

$$MaRSE_{diff.} = \sum_{i=v_{min}}^{v_{max}} \sum_{j=1}^{2n} C\left(v_i, t_j\right) * (C_{Tot})^j \tag{4}$$

The selection of the observation window (v) determines the effectiveness of brute-force attacks. If the observation window is too short, the adversaries might not find the hidden message because the hidden message took a longer interval than the observation window. If the observation window is too long, the possible combinations are enormous. As an example, if the message length is known to be a constant of 16 characters (packets), a one-week observation window will yield $C(403200, 16) \approx 2^{253}$ possibilities of message sequence. Since the set addresses are unknown, for each of those possibilities the attacker must guess the set addresses as formulated in (3).

IV. System Evaluation

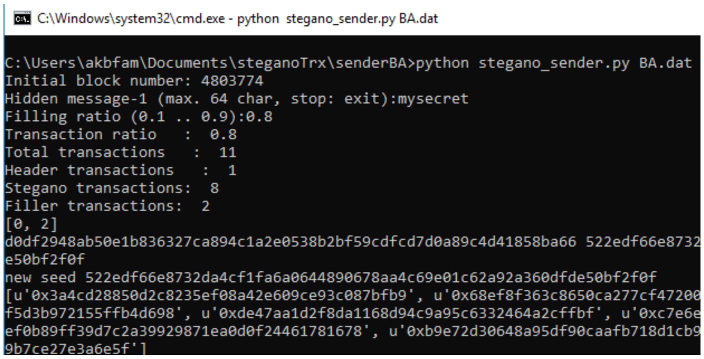
A. Implementation and Verifiability Test

We test the MaRSE method in Ethereum Rinkeby network by sending a hidden message. The configurations for the message delivery are presented in Table I. For the set address, we use 16 source addresses and 64 destination addresses. For each transaction, we randomly select 8 addresses as the subset each for both the source addresses and destination addresses. The configuration of the proper subset yields a message capacity of $2 * log_2 8 = 6$ bit per transaction. The message is sent with a constant interval of 15 seconds with additional sending and recording time to the blockchain network.

The demonstrations of the proposed method are shown by Fig. 3 - 5. Fig. 3 shows the sender is sending the hidden message. The receiver asynchronously scans the blockchain ledger for a new hidden message (Fig. 4). By default, the program will scan the ledger from the previously shared initial block number. However, the user can directly choose a different block number. In this evaluation, we start the scanning from block number 4932833 to speed up the scanning time. Once a new message is found, the sender will try to repeatedly scan 50 blocks at once to find the next valid stegano transaction. The process will be repeated

TABLE I
SYSTEM CONFIGURATION FOR EVALUATION

Parameter	Value
Initial block number	4803774
Hidden message	mysecret
Filling ratio	0.8
Total transactions	11
Header transaction	1
Stegano-transactions	8
Filler transactions	2
New initial block number	4932893
Initial scanning	4932833



C:\Windows\system32\cmd.exe - python stegano_sender.py BA.dat

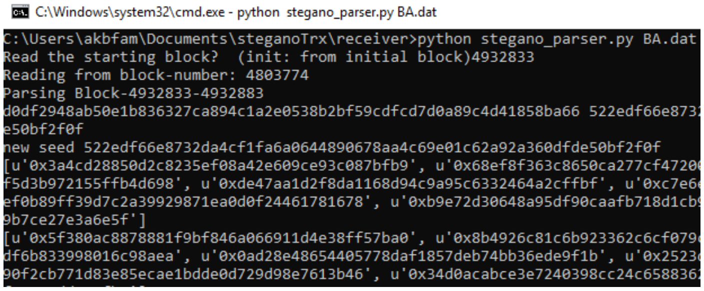
```
C:\Users\akbfam\Documents\steganoTrx\senderBA>python stegano_sender.py BA.dat
Initial block number: 4803774
Hidden message-1 (max. 64 char, stop: exit):mysecret
Filling ratio (0.1 .. 0.9):0.8
Transaction ratio   :  0.8
Total transactions   :  11
Header transactions   :  1
Stegano transactions:  8
Filler transactions:  2
[0, 2]
d0df2948ab50e1b836327ca894c1a2e0538b2bf59cdfcd7d0a89c4d41858ba66 522edf66e8732
e50bf2f0f
new seed 522edf66e8732da4cf1fa6a0644890678aa4c69e01c62a92a360dfde50bf2f0f
[u'0x3a4cd28850d2c8235ef08a42e609ce93c087bfb9', u'0x68ef8f363c8650ca277cf47200
f5d3b972155ffb4d698', u'0xde47aa1d2f8da1168d94c9a95c6332464a2cffbf', u'0xc7e6e
ef0b89ff39d7c2a39929871ea0d0f24461781678', u'0xb9e72d30648a95df90caafb718d1cb9
9b7ce27e3a6e5f']
```

Fig. 3. Sending hidden message

until all messages are received or waiting for a new block to be created (Fig. 5). The hidden message that was successfully recovered from the scanning process is ”mysecret”. This result proves the trusted verifiability of the secret channel. The sender can verify that the receiver definitely can scan the message as long as it was recorded in the blockchain. In another hand, the receiver cannot deny that the message has been sent in case it has been recorded in the ledger.

B. Message Delivery Covertness

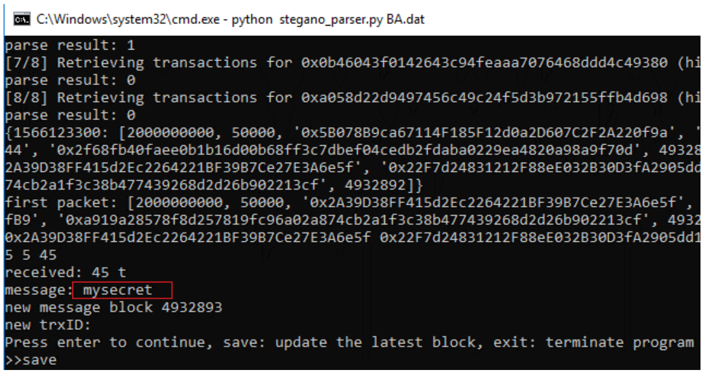
Compared to a single address schema, the MaRSE method that uses a multi-addresses schema has better secrecy. The number of stegano-transactions does not increase instantly when it starts sending the hidden message. In a single address schema, the generated transactions are noticeable for everyone. In this case, a particular address consecutively sends a large number of transactions. In our proposed method, the load to send the transaction is divided into 8 source addresses (S_S). It reduces the instant spike of the generated transactions and conceals the pattern from the attacker.



C:\Windows\system32\cmd.exe - python stegano_parser.py BA.dat

```
C:\Users\akbfam\Documents\steganoTrx\receiver>python stegano_parser.py BA.dat
Read the starting block?  (init: from initial block)4932833
Reading from block-number: 4803774
Parsing Block-4932833-4932883
d0df2948ab50e1b836327ca894c1a2e0538b2bf59cdfcd7d0a89c4d41858ba66 522edf66e873
e50bf2f0f
new seed 522edf66e8732da4cf1fa6a0644890678aa4c69e01c62a92a360dfde50bf2f0f
[u'0x3a4cd28850d2c8235ef08a42e609ce93c087bfb9', u'0x68ef8f363c8650ca277cf4720
f5d3b972155ffb4d698', u'0xde47aa1d2f8da1168d94c9a95c6332464a2cffbf', u'0xc7e6
ef0b89ff39d7c2a39929871ea0d0f24461781678', u'0xb9e72d30648a95df90caafb718d1cb
9b7ce27e3a6e5f']
[u'0x5f380ac8878881f9bf846a066911d4e38ff57ba0', u'0x8b4926c81c6b923362c6cf079
df6b833998016c98aea', u'0x0ad28e48654405778daf1857deb74bb36ede9f1b', u'0x2523
90f2cb771d83e85ecae1bdde0d729d98e7613b46', u'0x34d0acabce3e7240398cc24c658836
```

Fig. 4. Scanning for hidden message



C:\Windows\system32\cmd.exe - python stegano_parser.py BA.dat

```
parse result: 1
[7/8] Retrieving transactions for 0x0b46043f0142643c94feaaa7076468ddd4c49380 (hi
parse result: 0
[8/8] Retrieving transactions for 0xa058d22d9497456c49c24f5d3b972155ffb4d698 (hi
parse result: 0
{1566123300: [2000000000, 50000, '0x5B078B9ca67114F185F12d0a2D607C2F2A220f9a', '
44', '0x2f68fb40faee0b1b16d00b68ff3c7dbef04cedb2fdaba0229ea4820a98a9f70d', 49328
2A39D38FF415d2Ec2264221BF39B7Ce27E3A6e5f', '0x22F7d24831212F88eE032B30D3fA2905dd
74cb2a1f3c38b477439268d2d26b902213cf', 4932892]}
first packet: [2000000000, 50000, '0x2A39D38FF415d2Ec2264221BF39B7Ce27E3A6e5f',
fB9', '0xa919a28578f8d257819fc96a02a874cb2a1f3c38b477439268d2d26b902213cf', 4932
0x2A39D38FF415d2Ec2264221BF39B7Ce27E3A6e5f 0x22F7d24831212F88eE032B30D3fA2905dd1
5 5 45
received: 45 t
message: mysecret
new message block 4932893
new trxID:
Press enter to continue, save: update the latest block, exit: terminate program
>>save
```

Fig. 5. The extracted hidden message

The user can set different filling ratios (f_r) to further reduce the spiky number of transactions. The filling ratio enables a non-consecutive delivery that means the adversaries cannot just decode the generated transactions directly. The attackers must try all of the combinations of transaction sequences to retrieve the hidden message. If the set address of the sender is unknown, the attacker must first guess the correct possible combination.

In this paper, we limit our evaluations to empirical deployment into the public ledger. Further study to prove the robustness is to use stegano-analysis approach as presented in [11].

V. CONCLUSION

This paper proposed a secret and verifiable communication channel over blockchain transactions to ensure communication trustability between the communicating parties. It uses on-chain message delivery by using only multi-addresses without modifying the transaction fields for natural transaction patterns. The encoding schema (MaRSE) ensures the brute-force resistance features due to the sheer number of possible set addresses combinations and transaction sequences. The method adopts the filling ratio concept that inserts random transactions to further ensure the unpredictable pattern of the stegano-transactions. The system has been deployed to the Ethereum Rinkeby network and successfully transmitting and parsing the hidden message over the public ledger.

REFERENCES

[1] She, W., Huo, L., Tian, Z., Zhuang, Y., Niu, C., et al., ”A double steganography model combining blockchain and interplanetary file system.,” Peer-to-Peer Networking and Applications, 1-14, 2021.

[2] Mohsin, A. H., Zaidan, A. A., Zaidan, B. B., Mohammed, K. I., Albahri, O. S., et al., (2021). PSOBlockchain-based image steganography: towards a new method to secure updating and sharing COVID-19 data in decentralised hospitals intelligence architecture. Multimedia tools and applications, 80(9), 14137-14161.

[3] She, W., Huo, L., Tian, Z., Zhuang, Y., Niu, C., Liu, W. (2021). A double steganography model combining blockchain and interplanetary file system. Peer-to-Peer Networking and Applications, 1-14.

[4] Basuki, A. I., & Rosiyadi, D., ”Joint transaction-image steganography for high capacity covert communication,” In 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA) (pp. 41-46), IEEE, 2019

[5] Wang, W., & Su, C., ”Ccbrsn: a system with high embedding capacity for covert communication in bitcoin,” In IFIP International Conference on ICT Systems Security and Privacy Protection (pp. 324-337), Springer, Cham, 2020.

[6] Liu, S., Fang, Z., Gao, F., Koussainov, B., Zhang, Z., et al. ”Whispers on Ethereum: Blockchain-based Covert Data Embedding Schemes,” In Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure (pp. 171-179), 2020.

[7] Alsalami, N., & Zhang, B., ”Uncontrolled randomness in blockchains: Covert bulletin board for illicit activity,” In 2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS) (pp. 1-10), IEEE, 2020.

[8] R. Recabarren and B. Carbunar., ”Tithonus: A bitcoin-based censorship resilient system, Proceedings on Privacy Enhancing Technologies, vol. 2019, no. 1, pp. 68

[9] Partala, J.,”Provably Secure Covert Communication on Blockchain,” Cryptography, 2, 18, 2018.

[10] Dannen, C., ”Introducing Ethereum and solidity,” Vol. 318. Berkeley: Apress, 2017.

[11] Giron, A. A., Martina, J. E., & Custdio, R., ”Steganographic Analysis of Blockchains,” Sensors 21.12 (2021): 4078.