Journal of Network and Computer Applications 210 (2023) 103541

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

EBDL: Effective blockchain-based covert storage channel with dynamic labels

Can Zhang [a], Liehuang Zhu [a], Chang Xu [a,*], Zijian Zhang [a], Rongxing Lu [b]

[a] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China
[b] Faculty of Computer Science, University of New Brunswick, Canada

ARTICLE INFO

Keywords:
Blockchain
Covert channel
Covert communication

ABSTRACT

Blockchain-based covert communication has opened up a new research direction for the covert communication field. In blockchain-based storage covert communication, reliability can be guaranteed because of the inherent immutable property of blockchain. Besides, enhanced undetectability is also achieved because the sender and the receiver do not need to establish a direct connection or be online simultaneously. However, in existing blockchain-based solutions, pre-negotiated and fixed addresses are used as static labels to scan the specific transaction that embeds covert messages, which increases the risk of channel exposure. Moreover, some of them do not support large covert message transmission. In this paper, we present an effective blockchain-based covert storage channel with dynamic labels. More specifically, a novel dynamic label mechanism is proposed to help the receiver extract the covert message efficiently, whereas adversaries cannot distinguish whether a block contains specific transactions. Besides, a message segmentation mechanism is also proposed to achieve large covert message transmission. Theoretical security analysis and detailed experimental evaluations based on the public Ethereum blockchain network show that the proposed scheme is secure under Chosen Hidden-text Attacks with acceptable efficiency.

1. Introduction

With the development of computer networks, more and more people and enterprises use the Internet to communicate with each other. During the communication process, data including personal information or even business secret is transmitted via public channels, which brings out security and privacy issues. Hence, more and more privacy-preserving techniques based on cryptography have been proposed to achieve confidentiality and integrity of sensitive information during the communication processes. Unfortunately, in some scenarios, both the transmitted content and the communication behaviors need to be protected. For example, in the intelligence dissemination scenario, if the dissemination behaviors are revealed to adversaries, the relationship and communication purpose between the sender and the receiver will be exposed, and the spread of covert messages will be prevented.

To hide the communication behaviors on the Internet, the concept of covert communication (Lampson, 1973) is presented, which is based on the traditional steganography technique. In a typical covert communication scenario, a *sender* sends a covert message to a *receiver* through *covert channels*, where both the transmitted covert messages and the communication behaviors cannot be revealed to others. How to construct a covert channel under the public channel is a crucial part to achieve undetectable, reliable, and efficient covert communications.

Now researchers have proposed some covert channels based on network protocols and applications (Zhang et al., 2020; Liang et al., 2018a). However, in existing covert communication schemes, the sender must use designated addresses (e.g., IP addresses) related to both the sender and the receiver. Hence, if the address is leaked, their identities will be compromised. Besides, both the sender and the receiver need to be online during the whole process, especially in time-sensitive covert communication solutions. It also increases the risk of identity and behavior exposure.

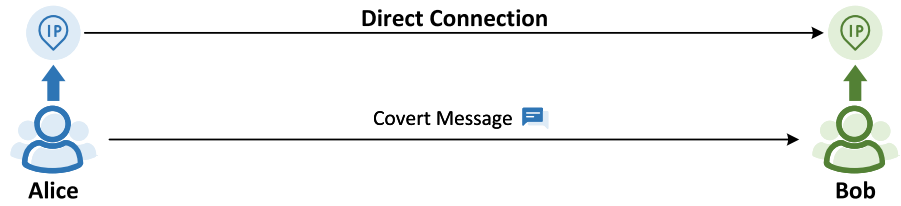
The proliferation of blockchain technology has attracted more and more researchers' attention. Hence, the blockchain-based covert channel model has been proposed. For example, assume a sender `Alice` wants to send a covert message to a receiver `Bob` via a blockchain-based covert channel. First, she embeds the covert message into a blockchain transaction, which will be packed into one of the newly-generated blocks with the help of blockchain miners. To recover the covert message, `Bob` synchronizes the newly-generated blocks, scans the specific transaction sent by `Alice`, and recovers the covert message from the transaction.

For covert communication, blockchain has two natural advantages. First, the average daily transaction counts of Bitcoin and Ethereum in the first half of 2022 are about 270,000 and 1,280,000, respectively.[1]

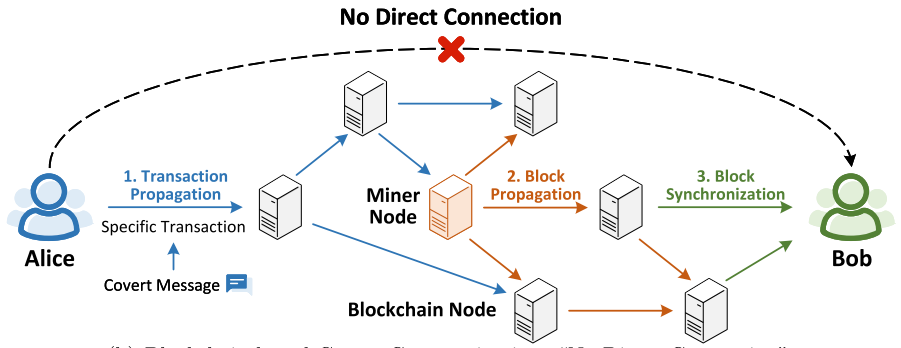
* Corresponding author.
E-mail address: xuchang@bit.edu.cn (C. Xu).

[1] Source: https://explorer.btc.com.

https://doi.org/10.1016/j.jnca.2022.103541
Received 28 January 2022; Received in revised form 19 October 2022; Accepted 29 October 2022
Available online 17 November 2022



(a) Traditional Covert Communication: "Direct Connection".



(b) Blockchain-based Covert Communication: "No Direct Connection".

Fig. 1. Difference between traditional and blockchain-based covert communication.

Hence, the specific transaction that embeds covert messages can be well hidden in a large volume of blockchain transactions. Second, transaction synchronization is based on broadcast communications instead of P2P communications. As shown in Fig. 1(a), in traditional covert communications, the sender `Alice` and the receiver `Bob` are directly connected, which means that their communication behaviors are easily exposed and their IP addresses are easy to be tracked. Besides, if one of them is offline, the communication process will be terminated, which brings out the issue of unreliability. As for blockchain-based covert communications, as shown in Fig. 1(b), `Alice` only needs to send the specific transactions to the blockchain, and `Bob` only needs to synchronize the newly-generated blocks and scan the specific transactions. These operations for both two sides are asynchronous. Hence, during the whole process, `Alice` does not need to be directly connected to `Bob`, which decreases the risk of being exposed. Therefore, blockchain can be considered as a natural medium to provide covert communications.

Although some blockchain-based covert communication mechanisms have been proposed in recent years, there remain some issues that should be solved. More specifically:

- **Realizability**: Some existing schemes are based on the ideal blockchain model. For example, the model used in the scheme in Partala (2018) assumes that "a new block appears after a finite and fixed amount of time", which is hard to be realized in real-world blockchain platforms.
- **Undetectability**: Most existing schemes are based on transaction scanning mechanisms under a fixed label, of which the undetectability cannot be guaranteed. For example, the sender uses a fixed blockchain address a to send specific transactions, which means that all the transactions with the input address a can be considered as specific transactions. If one of the specific transactions is leaked, adversaries can easily find the remaining specific transactions, and all the historical and future covert communication behaviors will be exposed.
- **Scalability**: Some schemes do not support large covert message transmission. Assume the sender wants to transmit a message with a length of 1 kB, he needs to split the message into multiple parts and send more than one specific transaction. Unfortunately, the transaction propagation mechanism cannot guarantee the consistency of the sending sequence of these specific transactions sent by the sender. Hence, the receiver cannot successfully recover the covert message although he can obtain all the specific transactions.

To tackle these issues, we design an efficient blockchain-based covert storage channel, named EBDL, that leverages the following two presented mechanisms: *Dynamic Label* and *Message Segmentation*. More specifically, the proposed dynamic label mechanism achieves efficient transaction scanning for the receiver. Besides, the proposed message segmentation mechanism enables the sender to send longer covert messages. To the best of our knowledge, EBDL is the first blockchain-based covert storage channel that uses the dynamic label to scan the specific transaction, and supports large covert message transmission simultaneously.

Following are the major contributions of this work:

- To achieve *Realizability*, we design a novel architecture of the blockchain-based covert channel, of which the corresponding threat model and Chosen Hidden-text Attacks (CHA) security model are also introduced in detail.
- To provide *Undetectability*, we propose a novel dynamic label mechanism that can be used in blockchain-based covert communications to help the receiver efficiently scan the specific transaction from large volumes of transactions. Meanwhile, adversaries cannot distinguish between specific transactions and normal transactions that do not store any kind of covert messages.
- To capture *Scalability*, we present a message segmentation mechanism that splits one covert message into multiple segments that are embedded in multiple specific transactions. Based on the presented message segmentation protocol, the receiver can successfully recover the segmented data sent by the sender.
- We make a thorough theoretical analysis and experimental evaluations based on Ethereum blockchain networks to prove that the proposed EBDL scheme achieves CHA-security, confidentiality, authentication, and reliability with acceptable efficiency.

The remaining paper is organized into the following sections. Section 2 introduces the related works about traditional covert communications and blockchain-based covert communications. Section 3 describes the fundamental knowledge of blockchain and the definition

of blockchain-based covert communications. The formalized system model, threat model, security model, and design goals are presented in Section 4. Section 5 gives a detailed description of the proposed EBDL scheme. Theoretical analysis and experimental evaluations are respectively shown in Sections 6 and 7. Finally, Section 8 concludes the paper.

2. Related works

In this section, we will introduce the related works about traditional covert communication and blockchain-based covert communication.

2.1. Traditional covert communication

Lampson (1973) first proposed the concept of the covert channel which is used for covert message communications where only the sender and the receiver know the communication behaviors. Now more and more network-based covert channels are proposed because of the proliferation of the Internet, which can be classified into two categories: the Covert Storage Channel (CSC) and the Covert Time Channel (CTC) (Association, 1985).

Existing covert communication schemes based on CSC leverage the package of network protocols (e.g., TCP/IP) to store covert messages. Rowland (1997) showed that weaknesses of TCP/IP protocol can be used by adversaries to form covert channels. For instance, adversaries can use the IP packet identification field, the initial SEQ field, and the ACK field of TCP to embed covert messages. Trabelsi et al. (2008) presented an ICMP-based covert channel based on record route IP header options. The proposed covert channel can be used to transfer covert messages and files. Some reservation fields of TCP/IP header can also be used to embed messages in CSC, whereas it is easy to be detected. Besides, the application-based CSC has also been proposed (Zhang et al., 2020), which achieves covert communication in various applications such as cloud computing and VoLTE.

The CTC can also be classified into two categories: the IPD-based CTC and packet-rearrangement CTC. More specifically, the IPD-based CTC uses the Inner Packet Delay (IPD) to embed covert messages, and the packet-rearrangement CTC uses the packet-rearrangement mechanism to store the covert messages. Gianvecchio and Wang (2011) summarized three typical IPD-based CTCs: Jitterbug, MB-CTC, and TR-CTC. For packet-rearrangement CTC, it can be constructed by VoIP traffics (Liang et al., 2018a,b) or VoLTE traffics (Zhang et al., 2018, 2019) in mobile networks. However, the reliability of CTC cannot be guaranteed because IPD and packet arrangement can be eliminated by adding random delay (Wang et al., 2009) or traffic shaping (Schulz et al., 2014), respectively.

In existing covert communication schemes, the sender needs to use the IP address and directly connects to the receiver during the covert communication process, especially in CTC-based covert communication scenarios. Under these circumstances, adversaries can trace the real identities of the sender and the receiver by the exposed IP addresses and covert communication behaviors.

2.2. Blockchain-based covert communication

Compared with traditional covert communications, blockchain-based covert communications achieve higher undetectability because of its large volume of transactions and the *"no direct connection"* property.

In recent years, more and more blockchain-based covert channels are proposed. Partala (2018) first proposed a provable secure covert communication scheme in blockchain, which is based on the ideal blockchain model and the CHA security model. Unfortunately, the ideal blockchain model is hard to be realized in the real-world blockchain platforms such as Bitcoin and Ethereum. Besides, the proposed scheme uses the Least-Significant Bit (LSB) of addresses to embed the covert message, which results in low channel capacities. Tian et al. (2019) proposed DLchain, a blockchain-based CSC that uses the Bitcoin ECDSA private key to embed covert messages and leverages dynamic labels based on the distribution of Bitcoin OP_RETURN field. However, the Bitcoin private key is leaked to the receiver, which may cause economical loss if the receiver is malicious. Fionov (2019) and Basuki and Rosiyadi (2019) explored the transaction field to find a suitable field for embedding covert messages in Bitcoin and Ethereum transactions, respectively. Gao et al. (2020) proposed a Bitcoin-based covert channel that achieves high undetectability on transaction scanning. However, the covert message is directly embedded in the OP_RETURN field, and the transaction scanning process is time-consuming. Cao et al. (2020) proposed a chain-based covert message embedding scheme in blockchain, where the transaction scanning is based on the address chain. However, the embedding scheme is based on the LSB. Liu et al. (2020) presented an Ethereum-based covert channel where the covert message is embedded in the transaction value field. Unfortunately, it does not support large covert message transmission.

In the recent two years, researchers have generated some new ideas in the field of blockchain-based covert communication. She et al. (2021) combined the traditional steganography on text/image and blockchain to construct a double steganography model, where Ethereum and IPFS are leveraged to realize the reliability of the covert message. Luo et al. (2022) leveraged the address interaction relationship and the transaction value field to embed the covert information. Zhang et al. (2022) constructed a voting contract and used the parameters to embed the covert message, and leveraged encryption algorithms to protect data privacy. However, those schemes use fixed (transaction or contract) addresses to scan the specific transaction, which increases the risk of being successfully detected. Zhang et al. (2021) used the field used by Ethereum's whisper protocol instead of transactions to construct a covert storage channel. Because the protocol package is not stored on the blockchain, after the expiration time is reached, the package cannot be obtained by the receiver. Liu et al. (2022) used Monero transaction amount to embed covert messages and achieve anti-detection. Unfortunately, if the sender tries to embed a large covert message in the transaction amount field, he will spend a lot of cryptocurrencies, which is unacceptable. Hence, it does not support large covert message transmission.

Based on the above analysis, to the best of our knowledge, none of the existing blockchain-based covert channels can provide *Realizability*, *Undetectability*, and *Scalability* simultaneously. Therefore, a practical blockchain-based covert channel should be presented, which forms the basic motivation of this work. In Section 7.4, we make a comparison between the proposed EBDL scheme and other relevant blockchain-based covert communication schemes. The result shown in Table 4 shows that among blockchain-based covert communication schemes, only EBDL can support the dynamic label and large covert message transmission.

3. Preliminaries

In this section, we will give the formal definition of blockchain and the blockchain-based covert communication scheme.

3.1. Blockchain

We first give the definition of blockchain that is based on the idea of the state machine, which will be used in the following descriptions of blockchain-based covert communications.

Definition 1 (*Blockchain from the State-machine Perspective*)**.** The blockchain can be abstracted as a state machine that consists of a current state S_n and a state update function Update. The state S_n consists of $n+1$ blocks $\{B_0, B_1, \ldots, B_n\}$ arranged in chronological order. More specifically, B_0 is the pre-defined genesis block, and for $i \in [1, n]$, B_i is the ith block that stores blockchain transactions. The block B_i can

be generated by any miner node in the blockchain network based on the consensus mechanism. When adding a block B_{n+1} to the blockchain, it is equivalent that each node in blockchain network executes the state update function $S_{n+1} \leftarrow \mathsf{Update}(S_n, B_{n+1})$ to update the stored blockchain data (i.e., the state of blockchain).

Note that in this definition we omit the detailed description of how B_i is generated, propagated, and verified. In the proposed EBDL scheme, we do not consider how to modify the existing blockchain networks because it is not practical to achieve covert communication in open channels.

3.2. Blockchain-based covert communication

Blockchain-based covert communication can be considered as a special kind of covert communication where it uses the blockchain networks instead of using traditional channels (e.g., VoIP covert channel Liang et al., 2018a) to carry the covert message. We give a formal definition of the proposed blockchain-based covert communication, which is based on the definition of traditional steganography (Hopper et al., 2002).

Definition 2 (*Blockchain-Based Covert Communication*)**.** A Blockchain-based Covert Communication (BCC) scheme $\Sigma = \{\mathsf{ParaGen}, \mathsf{Embed}, \mathsf{Extract}\}$ consists of the following three algorithms: The parameter generation algorithm ParaGen, and the covert message embedding/extraction algorithm Embed/Extract that work as follows:

- $param \leftarrow \mathsf{ParaGen}(\lambda)$ is a probabilistic algorithm. The input of the algorithm includes a security parameter λ. It outputs a covert communication parameter set $param$ that will be used for both the sender and the receiver to achieve covert communication.
- $S' \leftarrow \mathsf{Embed}(m, param, S)$ is a probabilistic algorithm. The input of the algorithm includes a covert message m that will be embedded in the blockchain transaction, the parameter set $param$ and the current state of blockchain S. It outputs the updated state of blockchain S'.
- $m \leftarrow \mathsf{Extract}(param, S'')$ is a deterministic algorithm. The input of the algorithm includes the parameter set $param$ and the current state of blockchain S''. It outputs the extracted covert message m.

Note that $S \subsetneq S' \subseteq S''$ are three different blockchain states. Assume the blockchain transaction T_s represents the specific transaction that stores the covert message m. Specifically, S represents the state before the sender sends T_s to the blockchain, S' denotes the state that T_s has been added to the blockchain, and S'' is the state that the receiver uses to extract the covert message stored in the specific transaction.

S' is the updated set of S because S and S' represent the input and output of the Embed algorithm, respectively. However, S' and S'' can be in the same state when the receiver synchronizes the state of the blockchain network (i.e., the block data) in real time.

4. Problem formalization

In this section, we will introduce the formal definition of the system model, the threat model, and the security model. Besides, we will give the design goals of the proposed scheme EBDL.

4.1. System model

A complete covert communication process consists of three process: *Embedding*, *Transmission* and *Extraction*, with three entities participated: the ***sender***, the ***receiver***, and the ***blockchain*** network, as shown in Fig. 2.

- *Embedding*: Like existing works in Tian et al. (2019), Gao et al. (2020), Cao et al. (2020) and Liu et al. (2020), we assume that before the covert communication process, the sender and the receiver have negotiated the shared key and parameters offline. During the embedding process, the ***sender*** constructs the covert messages and embeds them in specific blockchain transactions.
- *Transmission*: The ***blockchain*** has the characteristic of *decentralization* and *immutability* which can be considered as the medium to achieve covert communication. After generating specific blockchain transactions, the ***sender*** will send them to a blockchain node. Then, these specific transactions will be broadcast to other blockchain nodes, packed into one block with the help of the miner node, and finally stored on the blockchain.
- *Extraction*: During the extraction process, to get the covert message sent by the sender, the ***receiver*** needs to scan the corresponding specific blockchain transactions from thousands of transactions in each block and extracts the covert messages.

4.2. Threat model

From our perspective, public blockchain platforms such as Ethereum are more suitable for performing covert communication because of their large number of users and transactions. However, everyone can access the transaction data stored on the public blockchain because of its transparency property. Hence, the specific transactions are also exposed to the public like normal transactions.

- Passive adversaries try to periodically synchronize the block data, make statistical analysis, and try to infer which block contains specific transactions. If they find one of these specific transactions, both the covert messages and the blockchain addresses used by the sender will be leaked.
- Active adversaries try to impersonate the sender to send transactions that contain fake covert messages. This kind of attacks try to make the receiver extract fake covert messages and decrease the reliability of the proposed covert communication scheme.

Although active adversaries can also perform *Sybil Attacks* or *Eclipse Attacks* to undermine the availability of the blockchain network. If the attacks are successful, the sender's specific transaction cannot be added to the block, and the receiver cannot get the corresponding covert message. However, how to resist these kinds of attacks depends on the security of the blockchain itself, interested readers can refer to Tschorsch and Scheuermann (2016) and Ghosh et al. (2020) to get more relevant information. Hence, in our threat model, such kind of attacks are not considered.

4.3. Security model

Some of the existing blockchain-based covert communication schemes use the security model based on the Chosen Hidden-text Attacks (CHA) experiment (Partala, 2018; Alsalami and Zhang, 2018) which is widely used to prove the security and undetectability of hidden covert messages on traditional steganography (Dedic et al., 2009).

From our perspective, the aforementioned *"no direct connection"* property of blockchain-based covert communication means that the sender does not need to designate the receiver's (IP or blockchain) address during the transmission. Under this circumstance, the adversaries have to scan the specific transaction when synchronizing a block that contains thousands of transactions. Hence, we modify the existing CHA model based on the blockchain-based covert communications where the blockchain is considered as the state machine. When a block is added to the blockchain, the state is updated. In the modified CHA model, if all the Probabilistic Polynomial-Time (PPT) adversaries cannot distinguish whether the state update increment (i.e., the newly-added blocks) contains the specific transaction, the proposed scheme is proven to

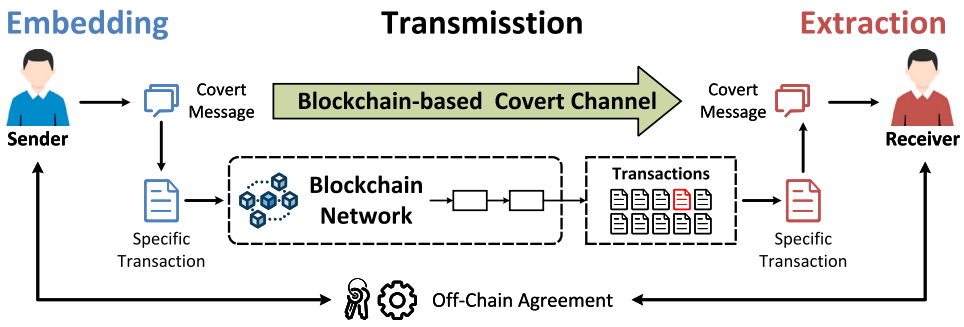


Fig. 2. System model.

be CHA-secure. Note that, similar to Partala (2018) and Alsalami and Zhang (2018), the CHA security model in BCC does not consider the active adversaries.

Definition 3 (*CHA Security in BCC*)**.** Let the scheme $\Sigma = \{\mathsf{ParaGen}, \mathsf{Embed}, \mathsf{Extract}\}$ be a BCC scheme, and S/S' be the current/updated state of blockchain. Using a security parameter λ, introducing a challenger $\mathcal{C}$, and an adversary $\mathcal{A}$, we design a probabilistic experiment $\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},\mathcal{C}}(1^{\lambda})$ as:

- The adversary $\mathcal{A}$ chooses a message m from the message space M and sends m to the challenger $\mathcal{C}$.
- The challenger $\mathcal{C}$ randomly chooses a bit $b \in \{0, 1\}$.
- If $b = 0$, $\mathcal{C}$ executes $param \leftarrow \mathsf{ParamGen}(\lambda)$ and $S' \leftarrow \mathsf{Embed}(m, param, S)$ to generate system parameters and a specific transaction where the covert message m is embedded, and gets the updated blockchain state S' (that includes the specific transaction). Otherwise, $\mathcal{C}$ sends a normal transaction and gets the updated blockchain state S' (that includes the normal transaction).
- $\mathcal{C}$ computes $\Delta = S' \setminus S$ which represents the block state update increment, and sends Δ to $\mathcal{A}$.
- $\mathcal{A}$ tries to distinguish whether Δ includes the specific transaction. Finally, $\mathcal{A}$ outputs a bit b' which represents the output of this experiment.

For all PPT adversaries $\mathcal{A}$, we define the advantage that $\mathcal{A}$ wins above experiment is:

$$\mathsf{Adv}^{\mathrm{CHA}}_{\mathcal{A},\Sigma}(1^{\lambda}) = |\Pr[\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},\mathcal{C}}(1^{\lambda})] - \frac{1}{2}|. \tag{1}$$

If $\mathsf{Adv}^{\mathrm{CHA}}_{\mathcal{A},\Sigma}(1^{\lambda})$ is negligible, we say the blockchain-based covert communication scheme Σ is CHA-secure.

4.4. Design goal

Based on the formal models and definitions given above, we introduce the design goals of the proposed scheme EBDL.

- *Undetectability*: The most important goal of covert communication is to achieve undetectability, which means passive adversaries are unable to detect the covert channel. Based on the presented security model, the proposed blockchain-based covert channel scheme should achieve CHA security.
- *Confidentiality & Authentication*: Different from traditional covert channels, in the blockchain-based covert channel, the data of specific transactions are stored on the blockchain publicly and permanently. Hence the confidentiality of the covert message should be guaranteed. Besides, authentication should also be guaranteed so that active adversaries cannot impersonate the sender to send fake covert messages.
- *Reliability*: The reliability of the covert communication scheme means that the receiver can successfully receive all the covert messages sent by the sender. More formally, the error rate and the recall loss during the covert communication process are negligible.
- *Efficiency*: The specific transaction scanning mechanism for the receiver should be efficient. More specifically, the scanning time of each block should not exceed the block generation time of the blockchain network (e.g., 12 s–15 s in Ethereum). If not, the receiver does not have enough time to scan the specific transactions. Besides, the scheme should support large covert message transmission, which forms one of the significant contributions of the proposed scheme.

In Section 7, we use *Kolmogorov–Smirnov (KS) Test* and *Kullback–Leibler Divergence (KLD) Test* to quantify the undetectability. More, specifically, KS test tries to find the difference in the feature distributions between specific transactions where covert messages are embedded, and normal transactions with no covert messages embedded. More specifically, the KS test tries to distinguish two probabilistic distributions F and G by taking the supremum of the absolute difference for all possible values of variable x, as shown in Eq. (2):

$$D_{KS} = \sup_{x} |F(x) - G(x)|. \tag{2}$$

In the KS test, the associated *p-value* is used to distinguish whether the distribution of two samples from F and G has significant differences, where $p > 0.05$ means that the two samples follow the same distribution.

KLD test, also named the Relative Entropy test, is also used to calculate the difference between two probabilistic distributions $F(x)$ and $G(x)$, as shown in Eq. (3):

$$D_{KL}(F \parallel G) = \sum_{x} p(x) \log(\frac{F(x)}{G(x)}). \tag{3}$$

If two random variables F and G follow the same distribution, $D_{KL}(F \parallel G) = 0$. Otherwise, if $D_{KL}(F \parallel G)$ increases, the distribution difference between F and G is larger.

5. The proposed EBDL scheme

In this section, we first give an overview about how the proposed EBDL scheme achieves undetectability and scalability. Then, we give a detailed description of the proposed scheme, which consists of three sub-processes: *Embedding*, *Transmission*, and *Extraction*.

5.1. Overview

As analyzed above, some of the existing schemes use a fixed label to scan specific transactions, so undetectability cannot be guaranteed.

Table 1
List of Notations.

Notation	Description
$\lambda = 2^k$	Security parameter.
K	The pre-negotiated key for label generation.
r	The pre-negotiated obfuscation parameter.
a_{in}	The input address of the specific transaction.
a_{out}	The output address of the specific transaction.
tag	An l-bit dynamic label.
F	A secure pseudo-random function.
L_s	The minimum length of each partition.

Besides, some of them do not support large covert data transmission, and cannot provide the scalability of blockchain-based covert communication.

To guarantee the *undetectability* of the blockchain-based covert communication, we propose a dynamic label mechanism, in which the fixed input address is not exploited. More specifically, in the *Specific Transaction Generation* sub-process, the sender uses one-time input and output addresses to generate the dynamic label and embeds it on the customized part of the blockchain transaction (e.g., OP_RETURN in Bitcoin and INPUT in Ethereum). In the *Specific Transaction Scanning* sub-process, when synchronizing newly-added blocks, the receiver scans the specific transactions which include the dynamic label.

Besides, to guarantee the *scalability* of the blockchain-based covert communication, we propose a covert message segmentation mechanism that can be used to realize large covert message transmission. More specifically, in the *Covert Message Embedding* sub-process, the sender segments, encrypts, and stores the split covert message separately in the customized data field of specific blockchain transactions. In the *Covert Message Extraction* sub-process, the receiver extracts the covert message from the scanned specific transactions.

5.2. Notations

Now we introduce the notations and the cryptographic primitives used in EBDL. The notations used in this section are illustrated in Table 1.

We use a secure pseudo-random function $F : \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^\lambda$ for all the PPT adversaries.

To encrypt the segmented covert message, we use a secure symmetric key cryptosystem SKE. More specifically, it includes two algorithms:

- Encryption: $c \leftarrow \mathsf{Enc}(K, m)$. It receives a secret key K and a message m as input, and it outputs a ciphertext c.
- Decryption: $m \leftarrow \mathsf{Dec}(K, c)$. It receives a secret key K and a ciphertext c as input, and it outputs the plaintext m.

Assume a bit-string $b = b_1 b_2 \cdots b_n$ with length $|b| = n$, $b[m] = b_1 b_2 \cdots b_m, m \le n$ represents the highest m-bits of b. We also use the symbols $\|$ and $\oplus$ to represent the concatenation and XOR operation respectively.

5.3. Embedding

In the *Embedding* process, the sender splits the covert message into n segments, and generates n specific transactions. Each segment and the corresponding dynamic label used for transaction scanning will be embedded in one of the specific transactions.

5.3.1. Covert Message Embedding

In the proposed scheme, we choose the customized data field to embed the covert message because it can be considered a designated data storage field in blockchain transactions. However, if the sender stores a large covert message in one specific transaction, this transaction might be suspected by adversaries. Hence, if he wants to transmit a long covert message, segmentation is necessary because it enhances the undetectability of specific transactions.

Although the sender and the receiver do not pre-negotiate the fixed input address, several pre-negotiated keys & parameters are also required. More specifically, the sender generates $K_l, K_e \leftarrow \{0,1\}^\lambda$ that will be used to generate the dynamic label for the blockchain and encrypt the segmented covert message, respectively. The obfuscation parameter $r \leftarrow \mathbb{Z}_{\lambda/2}$ used to compute the length of the label is also generated. We assume that both the sender and the receiver secretly store these shared keys & parameters.

Assume a sender wants to send a large covert message m to the receiver, which is partitioned into n parts, where n is an integer that satisfies $2 \le n \le 2^8$ (if $n = 1$, the message segmentation mechanism is not needed). First, the sender randomly chooses a partition $M = \{m_1, m_2, \ldots, m_n\}$ s.t. $m = m_1 \| m_2 \| \cdots \| m_n$, and $|m_i| \ge L_s, \forall i = 1, 2, \ldots, n$. The parameter L_s represents the minimum length of each partition. Then, the sender executes Algorithm 1 to construct n segments $\{ds_1, \ldots, ds_n\}$, where $ds_i (i \in [1, n])$ contains a partitioned covert message m_i.

Note that n is higher means the average length of segmentation is lower, which further enhances the security & undetectability of specific transactions. However, more specific transactions will be constructed by the sender and scanned by the receiver. Hence, how to choose n becomes a trade-off between security and efficiency. Here we omit the detailed description of the partition method for simplicity.

Algorithm 1 MsgSeg

Input: The number of partitions n, the covert message partition $M = \{m_1, m_2, \cdots, m_n\}$, an encryption key K_e.
Output: A set $S = \{ds_1, \cdots, ds_n\}$ that contains n data segments.
1: initialize S.
2: set $seq \leftarrow \mathbb{Z}_{2^{16}}$.
3: **for** $ctr = 1$ to n **do**
4: set $m' = seq||n||m_{ctr}$.
5: set $ds_{ctr} = \mathsf{SKE.Enc}(K_e, m')$.
6: add ds_{ctr} to S.
7: compute $ctr = ctr + 1$.
8: compute $seq = seq + 1 \mod 2^{16}$.
9: **end for**
10: **return** S.

According to the flooding mechanism in the blockchain instead of the propagation mechanism used in traditional covert P2P channels, the sequence of specific transactions received asynchronously by the receiver may not match the specific transaction sending sequence. Hence, a sequence number seq is also assigned to help the receiver rearrange the segmented data and extract the covert message correctly.

As shown in line 4 in Algorithm 1, we construct the segmentation protocol field $m' = seq\|n\|m_{ctr}$. Fig. 3 illustrates the designed message segmentation protocol field. Note that the sequence number seq and the partition number range from $[0, 2^{16} - 1]$ and $[0, 2^8 - 1]$, respectively. Hence, we set $n \in \mathbb{Z}^*_{2^8} \setminus \{1\}$, and set $seq \in \mathbb{Z}_{2^{16}}$.

Finally, the Algorithm 1 returns a set $S = \{ds_1, \ldots, ds_n\}$ that contains n data segments.

5.3.2. Specific Transaction Generation

After obtaining S, the sender generates n specific transactions. Each data segment $ds_i \in S$ is embedded in a specific transaction. Next, we will give a detailed description of how to construct these specific transactions.

First, the sender generates n normal transactions $\{T_1, \ldots, T_n\}$, which will be used to generate specific transactions and will not be sent to the

Field Name	Seq Number *seq*	Partition Number *n*	Segmented Message m_{ctr}
Size	16 Bits	8 Bits	$\geq L_s$ Bits
Range	0-65535	2-255	.....

Fig. 3. Message segmentation protocol field.

Algorithm 2 SpecTxGen

Input: A security parameter $\lambda = 2^k$, a set $S = \{ds_1, \cdots, ds_n\}$ that contains n data segments, a set $T = \{T_1, \cdots, T_n\}$ that contains n normal transactions, a label key K_l, and an obfuscation parameter r.
Output: A set $T' = \{T'_1, \cdots, T'_n\}$ that contains n specific transactions.
1: initialize T'.
2: set $k' = k - 1$.
3: **for** $i = 1$ to n **do**
4: parse T_i as (a_{in}, a_{out}, D).
5: compute $\rho = a_{out}[k'] \oplus r$.
6: compute $l = \rho + \lambda/2 + 1$.
7: calculate $tag = F(K_l, a_{in})[l]$.
8: set $D' = tag||ds_i$.
9: set $T'_i = (a_{in}, a_{out}, D')$.
10: add T'_i to T'.
11: **end for**
12: **return** T'.

Algorithm 3 SpecTxScan

Input: A security parameter $\lambda = 2^k$, a set $B = \{B_1, B_2, \cdots\}$ that contains the receiver's newly-synchronized blocks, a label key K_l, and an obfuscation parameter r.
Output: A specific transaction list L.
1: initialize L.
2: set $k' = k - 1$.
3: **for each** block $B_i \in B$ **do**
4: **for each** transaction $T \in B_i$ **do**
5: parse T as (a_{in}, a_{out}, D).
6: compute $\rho = a_{out}[k'] \oplus r$.
7: compute $l = \rho + \lambda/2 + 1$.
8: calculate $tag = F(K_l, a_{in})[l]$.
9: set $tag' = D[l]$.
10: **if** $tag == tag'$ **then**
11: add T to L.
12: **end if**
13: **end for**
14: **end for**
15: **return** L.

blockchain. Then, as shown in line 3–11 in Algorithm 2, for each data segment ds_i that contains the covert message segment m_i, the algorithm chooses a normal transaction $T_i = (a_{in}, a_{out}, D)$ from T. For $i \in [1, n]$, T_i contains a (one-time) input address a_{in}, a (one-time) output address a_{out}, and a customized data field D (i.e., the OP_RETURN or INPUT field), and at that time, $D = \perp$.

Next, the Algorithm 2 computes the dynamic label tag used to scan T_i. More specifically, to compute the dynamic label, the algorithm first chooses the highest k' bits of the output addresses a_{out}. Then it executes an XOR operation to generate ρ, and sets $\rho+\lambda/2+1$ as the length l. Note that we use $\rho+\lambda/2+1$ instead of ρ to represent the label's length because the label tag is generated by the pseudo-random function F, and the probability of collision is 2^{-l} where l represents the label's length. When l is smaller, the collision probability is higher, which means that the reliability of covert communication cannot be guaranteed.

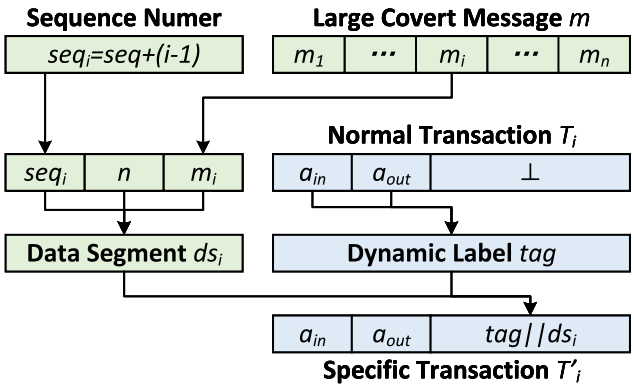


Fig. 4. An example of the Specific Transaction T'_i.

The Specific Transaction T_i

Input Address	1110****0010	Used to generate dynamic label
Output Address	0100100****	The length is *L=(36⊕r+129)* bits
Customized Data Field	0x2c38****	*L*-bit Dynamic Label
		The *i*-th Data Segmentation ds_i

Fig. 5. An example of the Specific Transaction T'_i.

Then, it sets $D' = tag \parallel ds_i$, and generates a new transaction $T'_i = (a_{in}, a_{out}, D')$, which can be considered as the corresponding specific transaction associated with T_i and ds_i.

Fig. 4 gives an example of the specific transaction T'_i. The green part represents the process of the *MsgSeg* algorithm that outputs the ith data segment ds_i. The blue part represents the process of the *SpecTxGen* algorithm that generates the corresponding specific transaction T'_i, of which the customized data field contains the dynamic label tag and the data segment ds_i.

Finally, the Algorithm 2 returns a set T' that contains all the specific transactions, which will be sent to the blockchain.

We give an example to explain how the specific transaction is constructed in Fig. 5. Specifically, in the specific transaction T_i associated with the data segment ds_i, the output address $a_{out} = 0100100\cdots$, the highest 7 bits are 0100100 (36 Decimal). Hence, the length of label is $l = 36 \oplus r + 129$ bits. The customized data field stores the l-bit label and the encrypted data segment ds_i.

Note that a_{in} and a_{out} are one-time addresses randomly generated by the blockchain address generation mechanism. Hence, the length l ranges from $[\lambda/2 + 1, \lambda]$ also follows the uniform distribution. The detailed proof is given in Section 6.

5.4. Transmission

In the *Transmission* process, these n specific transactions will be sent to the blockchain and stored on the newly-generated blocks with the help of the miner node. The specific transactions are indistinguishable from normal transactions (which will be proven in Section 6). Hence, they will be packed in multiple blocks by miners and stored on the blockchain, just like normal transactions.

Besides, as mentioned above, during the transmission process, no direct connection between the sender and the receiver is needed. Hence, the senders do not need to make a TCP/IP connection to the receiver like traditional covert communications. However, to hide the transaction relationship between the sender and the receiver, the senders cannot designate the transaction output address as the receiver's address. Hence, the transaction scanning mechanism becomes a crucial part for the receiver. Otherwise, the receiver cannot identify which transaction is the specific transaction.

5.5. Extraction

In the *Extraction* process, the receiver scans all the specific transactions, and finally extracts the covert message.

5.5.1. Specific Transaction Scanning

In the receiver's view, he does not know the exact time that the sender sends the specific transaction. Hence, after synchronizing newly-added blocks $B = \{B_1, B_2, \ldots\}$, he executes Algorithm 3 to scan all the specific transactions stored in B.

In Algorithm 3, for each transaction $T \in B_i$ in each block $B_i \in B$, the algorithm computes a label tag based on the transaction's addresses and the secret key & obfuscate parameter. If the highest l bits of the customized data field is equal to the generated label (i.e., $tag = D[l]$), which means the transaction T can be considered as a specific transaction that will be added to the specific transaction list L. Finally, the receiver obtains L.

To further improve the scanning efficiency, the receiver can record a variable h, which represents the highest block height that the receiver has executed the transaction scanning operation. Hence for the next covert communication round, the receiver can start with the block height h instead of the genesis block. Here, we omit the detailed description for simplicity.

5.5.2. Covert Message Extraction

After the transaction scanning process, the receiver obtains n specific transactions $\{T_1, \ldots, T_n\}$. Note that n is only used as a symbolic representation, it does not mean that the receiver knows n in advance. n will be known only if the receiver obtains & parses the first scanned specific transaction.

Algorithm 4 MsgExtract

Input: An encryption key K_e, and a specific transaction list $L = \{T_1, \cdots, T_n\}$.
Output: The extracted covert message m.
1: initialize tmp.
2: initialize a counter $ctr = 1$.
3: **for each** $T \in L$ **do**
4: parse T as $(addr^i, addr^o, D)$.
5: parse D as $tag||c$ based on the transaction scanning rule of Algorithm 3.
6: set $m'_{ctr} = \mathsf{SKE.Dec}(K_e, c)$.
7: parse m'_{ctr} as $seq_{ctr}||m_{ctr}$.
8: add (seq_{ctr}, m_{ctr}) to tmp.
9: compute $ctr = ctr + 1$.
10: **end for**
11: rearrange the entries in tmp in ascending order of seq.
12: /* assume $tmp = \{(seq_1, m_1), \cdots, (seq_n, m_n)\}$ */
13: set $m = m_1||m_2|| \cdots ||m_n$.
14: **return** m.

Then the receiver executes Algorithm 4 to extract the covert message m, which can be considered an inverse of the Algorithm 1. Note that the sequence number seq is used to rearrange the segmented data, as shown in line 11.

5.6. A complete process of EBDL

Now we give a sequence chart of a complete blockchain-based covert communication process of EBDL, as illustrated in Fig. 6. We believe that it can help the readers understand the proposed scheme.

- Before starting the blockchain-based covert communication, the ***sender*** and the ***receiver*** need to negotiate an encryption key K_e, a label key K_l and an obfuscation parameter r.
- In the *Embedding* process, the ***sender*** partitions the covert message m into n parts $M = \{m_1, \ldots, m_n\}$. Next, he gets a set of n data segments $S = \{ds_1, \ldots, ds_n\}$ by executing $MsgSeg(n, M, K_e)$. Then, he generates n normal transactions $T = \{T_1, \ldots, T_n\}$, and generates n specific transactions $T' = \{T'_1, \ldots, T'_n\}$ by executing $SpecTxGen(\lambda, S, T, K_l, r)$.
- In the *Transmission* process, the above n specific transactions $\{T'_1, \ldots, T'_n\}$ will be sent and broadcast to the ***blockchain***. With the help of blockchain miners, these transactions will be added to newly-generated blocks and stored on the ***blockchain***.
- In the *Extraction* process, when synchronizing several new blocks $B = \{B_1, B_2, \ldots\}$, the ***receiver*** obtains a list L that contains all the scanned specific transactions by executing $SpecTxScan(\lambda, B, K_l, r)$. Finally, he extracts the covert message m by executing $MsgExtract(K_e, l)$.

6. Theoretical analysis

In this section, we will make a theoretical analysis to prove that the proposed EBDL scheme achieves the undetectability, confidentiality, authentication, and reliability that are presented in the design goals.

6.1. Undetectability

First we make a formalized definition of the proposed scheme EBDL = {ParaGen, Embed, Extract} to adapt the modified CHA security model. Then we make sketch proof to prove that EBDL achieves CHA security.

6.1.1. Definition

Based on the CHA security model, EBDL can be defined as the combination of the following three algorithms:

- $param \leftarrow \mathsf{ParaGen}(\lambda)$: This algorithm is based on the key & parameter negotiation process. More specifically, the sender generates $param = \{K_e, K_l, r\}$ as the output of this algorithm, which will be shared with the receiver through secure channels.
- $S' \leftarrow \mathsf{Embed}(m, param, S)$: This algorithm is associated with Algorithm 1 and Algorithm 2 for the sender. More specifically, the sender first executes Algorithm 1 to generate n data segments of the large covert message m. Then, these data segments will be embedded on the corresponding n specific transactions, which are generated by executing Algorithm 2. Each of the specific transactions is associated with a dynamic label tag. We assume S and S' represent the state of the blockchain before and after the sender sends specific transactions, respectively.
- $m \leftarrow \mathsf{Extract}(param, S'')$: This algorithm is related to Algorithm 3 and Algorithm 4 for the receiver. More specifically, the receiver first executes the Algorithm 3 to scan the specific transactions. Then he executes Algorithm 4 to extract the segmented & encrypted covert message. Finally, the covert message m will be extracted as the output of this algorithm.

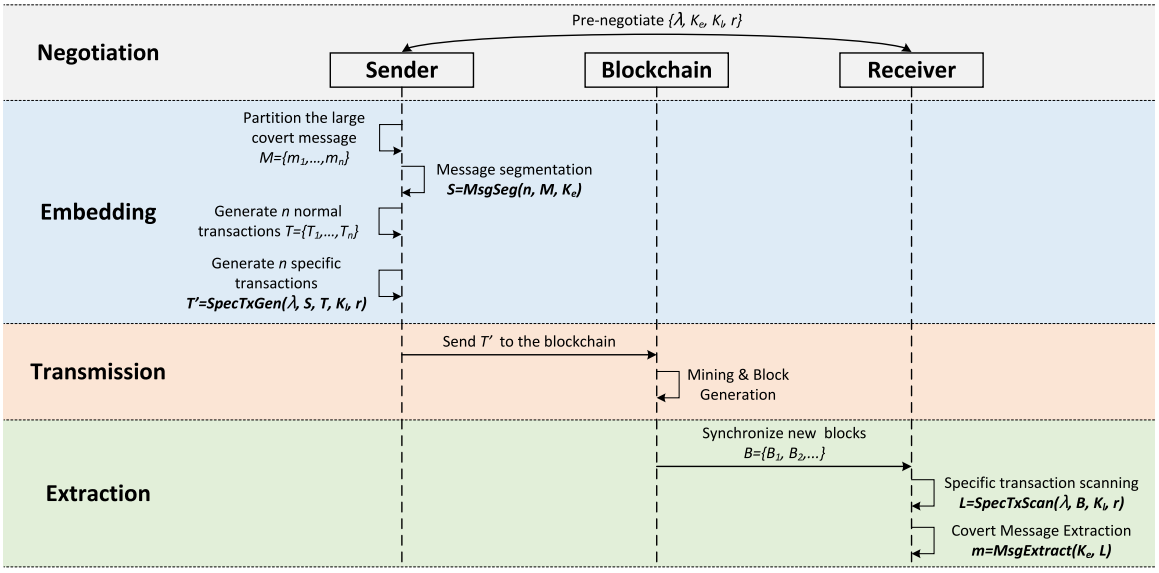


Fig. 6. The sequence chart of a complete process of EBDL.

6.1.2. Security proof

Now we prove that EBDL achieves CHA security. First, we introduce two lemmas used for security proof of the proposed scheme.

Lemma 1 (*Transaction Indistinguishability*). *If the cryptographic primitive F and* SKE *is secure, the specific transaction constructed by* Embed *Algorithm is indistinguishable from normal transactions for any PPT adversary $\mathcal{A}$.*

Proof of Lemma 1. In Algorithm 2, we assume that n normal transactions $T = \{T_1, \ldots, T_n\}$ are generated by the sender as the input. After executing Algorithm 1 and Algorithm 2, n specific transactions $T' = \{T'_1, \ldots, T'_n\}$ are generated. For each specific transaction $T'_i \in T'$ associated with the normal transaction $T_i \in T$, only the customized data field is different. Hence, we only need to prove that the customized data field cannot be used by $\mathcal{A}$ to scan specific transactions.

If the pseudo-random function F and the cryptosystem SKE are secure, the concatenation of the generated label tag and the cyphertext c is indistinguishable from a random string embedded in the customized data field in normal transactions. Without knowing K and r, $\mathcal{A}$ cannot scan specific transactions. Hence, the specific transactions that embed the dynamic label and the segmented ciphertext are indistinguishable from normal transactions for $\mathcal{A}$. □

To achieve enhanced indistinguishability, more complicated obfuscation mechanisms based on advanced statistical analysis should be designed. We leave it as future works.

Lemma 2 (*State Indistinguishability*). *Assume Δ_0 represents the block state update increment (i.e., newly-generated blocks) when C chooses $b = 0$ in $\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},C}(1^\lambda)$, which contains specific transactions. Assume Δ_1 represents the block state update increment when C chooses $b = 1$ in $\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},C}(1^\lambda)$, which does not contain any specific transaction. If the specific transaction is indistinguishable from normal transactions for any PPT adversary $\mathcal{A}$, Δ_0 and Δ_1 are also indistinguishable for $\mathcal{A}$*

Proof of Lemma 2. The only way that $\mathcal{A}$ can distinguish between Δ_0 and Δ_1 is to find which block contains specific transactions. However, if $\mathcal{A}$ cannot distinguish between specific transactions and normal transactions, he cannot find either Δ_0 or Δ_1 includes specific transactions. Therefore, Δ_0 and Δ_1 are indistinguishable for $\mathcal{A}$. □

Now we prove that the proposed EBDL scheme achieves CHA security based on the two lemmas mentioned above.

Theorem 1 (*CHA Security of EBDL*). *If the cryptographic primitive F and* SKE *are secure, the proposed scheme* EBDL = {ParaGen, Embed, Extract} *is secure under CHA attacks.*

Proof of Theorem 1. Based on the proof of Lemmas 1 and 2, we can conclude that if F and SKE are secure, $\mathcal{A}$ cannot distinguish between Δ_0 and Δ_1. Hence, during the experiment $\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},C}(1^\lambda)$, after receiving Δ from the challenger C, $\mathcal{A}$ cannot infer whether Δ contains specific transactions. Under that situation, $\mathcal{A}$ can only randomly guess the output b' with the probability $\Pr[\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},C}(1^\lambda)] = \frac{1}{2}$, and the advantage of $\mathcal{A}$

$$\mathrm{Adv}^{\mathrm{CHA}}_{\mathcal{A},\Sigma}(1^\lambda) = |\Pr[\mathbf{Expt}^{\mathrm{CHA}}_{\mathcal{A},C}(1^\lambda)] - \frac{1}{2}| \tag{4}$$

is negligible. Therefore, the proposed scheme EBDL = {ParaGen, Embed, Extract} is secure under CHA attacks. □

6.1.3. Unlinkability

Unlinkability of the specific transactions means that adversaries cannot link two different specific transactions sent by the same sender. Assume one specific transaction T'_0 is leaked to an adversary $\mathcal{A}$. If $\mathcal{A}$ can link another specific transaction T'_1 sent by the same sender, he will easily distinguish between specific transactions and normal transactions. Based on the Lemmas 1 and 2, the CHA security cannot be guaranteed because $\mathcal{A}$ can easily distinguish whether the newly-add blocks contain specific transactions.

Without loss of generality, we take a blockchain-based covert communication scheme Σ = {ParaGen, Embed, Extract} that uses a fixed input address a to scan specific transactions as an example. Assume two specific transactions $T'_0 = (a, a^0_{out}, D^0)$ and $T'_1 = (a, a^1_{out}, D^1)$ are sent by the sender. If T'_0 is leaked to an adversary $\mathcal{A}$, T'_1 will be easily linked by $\mathcal{A}$ because T'_0 and T'_1 have the same input address a. Hence, although the fixed label is efficient for the receiver's transaction scanning, security cannot be guaranteed. More specifically, all the previous specific transactions will be scanned by $\mathcal{A}$, and all the historical and future covert communication behaviors will be exposed.

Different from the above blockchain-based covert communication schemes, the proposed EBDL scheme achieves specific transaction unlinkability. Assume two specific transactions $T'_0 = (a^0_{in}, a^0_{out}, D^0)$ and $T'_1 = (a^1_{in}, a^1_{out}, D^1)$ is sent by the sender. If T'_0 is leaked to an adversary $\mathcal{A}$, T'_1 will not be linked because the four addresses $a^0_{in}, a^0_{out}, a^1_{in}, a^1_{out}$ are different one-time addresses, and the two customized data field D^0 and D^1 is indistinguishable from random strings if F and SKE are secure.

Compared with those fixed label-based schemes, in EBDL, we present a dynamic label mechanism for the receiver to scan specific transactions, which enhances the undetectability of the proposed scheme.

6.2. Confidentiality & Authentication

Confidentiality means that the covert message stored in the specific transactions cannot be recovered, and authentication means that adversaries cannot impersonate the sender to send specific transactions. If the pseudo-random function F and the SKE cryptosystem are secure, the proposed scheme EBDL will achieve confidentiality and authentication. Specifically, in specific transactions, the covert message is encrypted by SKE, of which the encryption key K_e is a pre-negotiated secret key. If SKE is secure, without knowing K_e, adversaries cannot recover the encrypted covert message.

Besides, the receiver uses the dynamic label $tag = F(K_l, a_{in})[l]$ generated by the sender to scan the specific transactions. If F is secure, without knowing the secret key K_l and the obfuscation parameter r, the corresponding label cannot be correctly generated. Hence, although active adversaries may try to impersonate the sender to send fake specific transactions, with the wrong dynamic label, these fake transactions cannot be scanned by the receiver.

6.3. Reliability

Because the proposed scheme can be considered as a blockchain-based covert storage channel, the covert message is stored in blockchain transactions. Due to the reliability of the blockchain, the block data cannot be modified or deleted which means the covert message cannot be tampered with.

During the covert communication process, the receiver uses the dynamic label generated by the pseudo-random function F. Considered that collision might occur in F, which can cause the receiver to misidentify normal transactions as specific transactions (i.e., False-Positive). Now we prove that the probability of FP is negligible.

Theorem 2. *If the pseudo-random function F and hash functions used in the blockchain are secure, r is uniformly sampled from $\mathbb{Z}_{\lambda/2}$, the probability* $\Pr[\text{FP}] \leq negl(\lambda)$ *where $negl(\lambda)$ is a negligible function of λ.*

Proof of Theorem 2. If the hash functions (i.e., SHA256 used in Ethereum) are secure, which means the generated addresses follow a uniform distribution. At the same time, r is also uniformly sampled from $\mathbb{Z}_{\lambda/2}$. Hence, $l = a_{out}[k'] \oplus r + \lambda/2 + 1$ also follows uniform distribution, which means the probability of taking any number in the integer set $[\lambda/2+1, \lambda]$ is $2/\lambda$. If F is secure, the output of F also follows the uniform distribution, which means the probability of collision of tag with length l is $1/2^l$. Combining the above equations, the probability $\Pr[\text{FP}]$ can be calculated as:

$$\begin{aligned}\Pr[\text{FP}] &= \sum_{i=\lambda/2+1}^{\lambda} \frac{2}{\lambda} \cdot \frac{1}{2^i} = \frac{2}{\lambda} \cdot (2^{-\frac{\lambda}{2}} - r^{-\lambda}) \\ &\leq \frac{\lambda}{2} \cdot 2^{-\frac{\lambda}{2}} = negl(\lambda),\end{aligned} \tag{5}$$

where $negl(\lambda)$ is a negligible function of λ. □

When $\lambda = 256$, the probability that the receiver misidentifies a normal transaction as a specific transaction is less than 2.3×10^{-41}, which is negligible. Hence, the reliability of the proposed EBDL scheme can be guaranteed.

7. Performance evaluation

In this section, we will make detailed experimental evaluations based on Ethereum (ETH) blockchain platform to show that the proposed EBDL scheme achieves undetectability with acceptable efficiency.

7.1. Experimental environment

To prove that the proposed EBDL scheme can be used on the public blockchain platforms, we deployed an ETH client on a remote Alibaba Cloud server running CentOS 7 x64 operating system with Intel CPU, 16 GB RAM, and 5Mbps network bandwidth. We also implement the proof-of-concept of the EBDL scheme written in Python 3 language, and we also use the SciPy library[2] to make the statistical analysis of both normal and specific blocks in the undetectability evaluation process.

Theoretically, to enhance the undetectability, both the sender and the receiver need to deploy an ETH full node and use them to send/receive specific transactions. However, due to the storage limit of our cloud server, we use the API provided by Etherscan[3] to send, receive, and query ETH blockchain and transaction data.

Our experimental evaluations are based on two blockchain networks: *Ethereum Mainnet* (ETH-MAIN), and *ETH Rinkeby Testnet* (ETH-TEST). We simulate the complete blockchain-based covert communication process on these networks and evaluate the undetectability and efficiency of the proposed EBDL scheme. In ETH, the segmentation mechanism is considered because the customized data field (i.e., the INPUT field) in ETH supports longer data storage. We manually set the length of the embedded covert message is 512 bit for each segment in specific ETH transactions.

7.2. Undetectability

In Section 6, a theoretical security analysis has proven that EBDL achieves CHA security. Now we use statistical analysis to further prove that adversaries cannot distinguish between the specific ETH block and the normal one.

The current covert channel detection methods mainly focus on trying to extract information that can identify a change in the distribution of a certain behavior feature or statistical anomaly (Iglesias and Zseby, 2017; Wang et al., 2017). In traditional covert channels, two types of detection metrics are often used to quantify the undetectability: *KS Test* and *KLD Test* that corresponds to Eqs. (2) and (3), respectively.

For each type of blockchain network, we send 9 specific transactions and obtain 9 specific blocks. For comparison, we also choose 9 normal blocks that do not contain any specific transactions. To eliminate the time influence, all normal blocks are chosen beside the specific blocks. For example, if the height of a specific block is h, we will choose the block with $h-1$ or $h+1$ as a normal block if it does not contain any specific transactions. Finally, for each blockchain network, we get 9 normal blocks N01 , …, N09 as the NORM dataset and 9 specific blocks S01 , …, S09 as the SPEC dataset.

For undetectability evaluation, we will prove that adversaries cannot distinguish between the specific blocks and the normal blocks. We assume that all the transaction fields except the customized data fields (i.e., INPUT in ETH) are indistinguishable from those of normal transactions. Hence, we focus on the statistical analysis of customized data fields. To further quantify the distribution difference, we use *KS Test* and *KLD Test* to metric the difference of character distribution between NORM and SPEC dataset for each blockchain network.

The result of *KS-Test* is shown in Fig. 7. As can be seen from Figs. 7(a) and 7(b), for both ETH-TEST and ETH-MAIN, all the KS p-values for normal blocks are greater than 0.05, which means that we can use *KS-Test* to evaluate the undetectability. As we can see from Figs. 7(c) and 7(d), all the KS p-values for specific blocks are also larger than 0.05, which means that *KS-Test* cannot distinguish the character distribution between NORM and SPEC for all the two blockchain networks.

[2] https://www.scipy.org.
[3] https://cn.etherscan.com/apis.

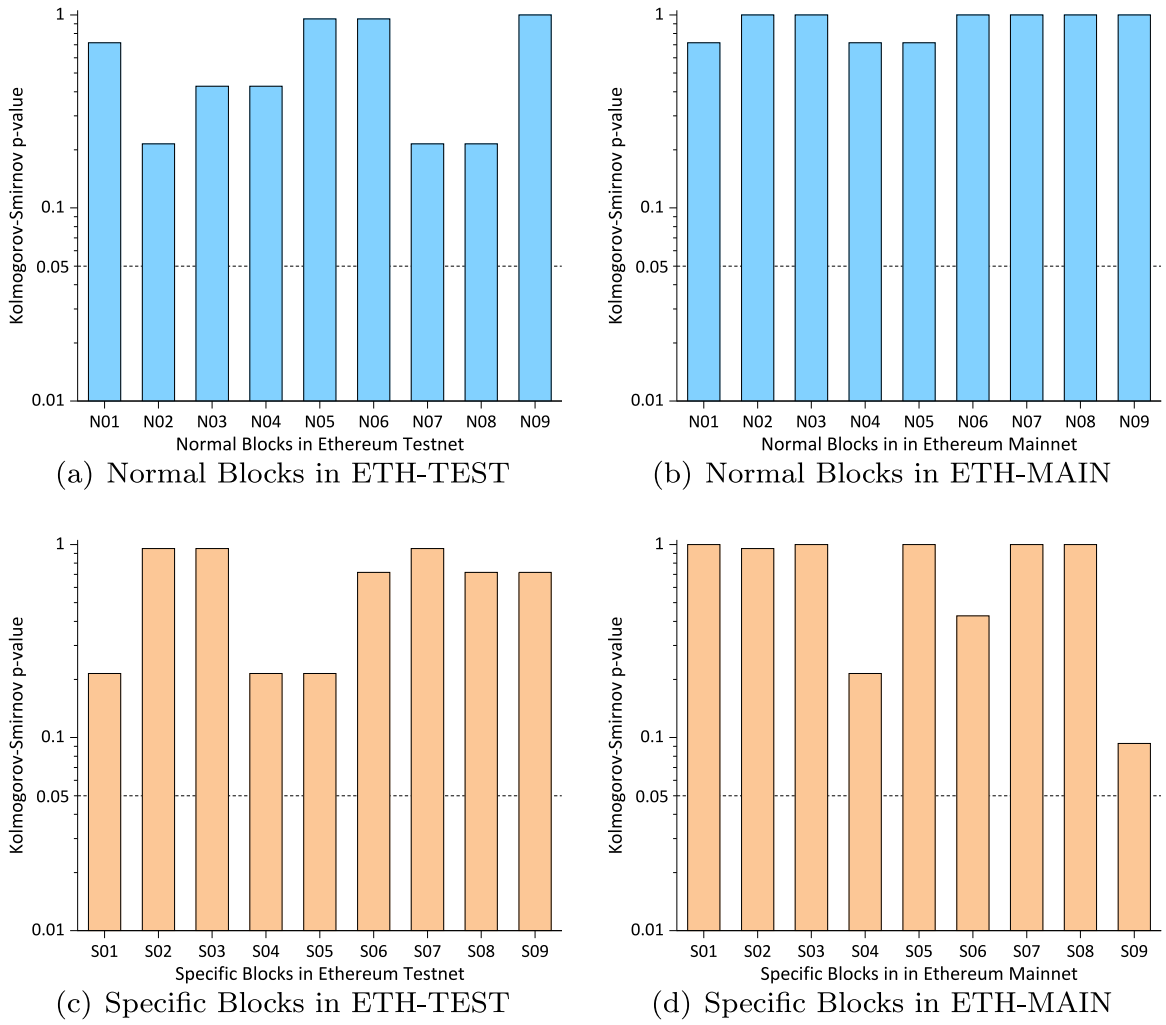


Fig. 7. Result of KS-Test.

The result of *KLD-Test* is shown in Fig. 8. Based on the results from Figs. 8(a) and 8(b) which represent the KLD value for normal blocks, we set the KLD threshold $\theta_{eth-test} = 0.01, \theta_{eth-main} = 0.01$. Based on these thresholds for each network, we can use *KLD-Test* to prove the undetectability of the proposed EBDL scheme. As is illustrated for Figs. 8(c) and 8(d), all the KLD values for specific blocks in different networks are lower than the corresponding threshold, which means that the *KLD-Test* cannot distinguish the character distribution between NORM and SPEC for all the four blockchain networks.

Based on the evaluation of both *KS-Test* and *KLD-Test*, we can conclude that the blockchain-based covert channel constructed by the proposed EBDL scheme is undetectable.

7.3. Efficiency

Now we discuss the efficiency of EBDL, which consists of the efficiency of the transaction scanning process and the complete covert communication process.

7.3.1. Transaction scanning process

For each blockchain network, we have obtained 9 normal blocks and 9 specific blocks. As the receiver, it needs to scan the specific transactions from all the newly-synchronized blocks. During the scanning process, the receiver will calculate the label for all transactions in each block and find the specific transaction where the calculated label has successfully matched the label stored in the transaction data. Hence, for each block, the computation complexity of transaction scanning is $O(n_t)$ where n_t represents the average number of transactions per block.

Table 2
Transaction scanning time per block.

Network type	Average scanning time (ms)
ETH-TEST	0.455
ETH-MAIN	2.994

We also evaluate the transaction scanning time per block, the experimental results are listed in Table 2. We can find that the scanning time for MainNet is longer than that of TestNet because MainNet has more transactions per block compared with TestNet. The result also shows that the transaction scanning process achieves acceptable efficiency because the transaction scanning time is much smaller than that of the block generation time for the ETH blockchain network.

7.3.2. Complete covert communication process

Now we simulate a complete covert communication process under a different number of segments. For both TestNet and MainNet, we set the number of segments (i.e., SegNum) that ranges from 1 to 5 with the increment of 1, which equals the number of transactions. We set the bit-length of the embedded covert message for each transaction as 512 bit. All the experiments are performed two times and we calculate

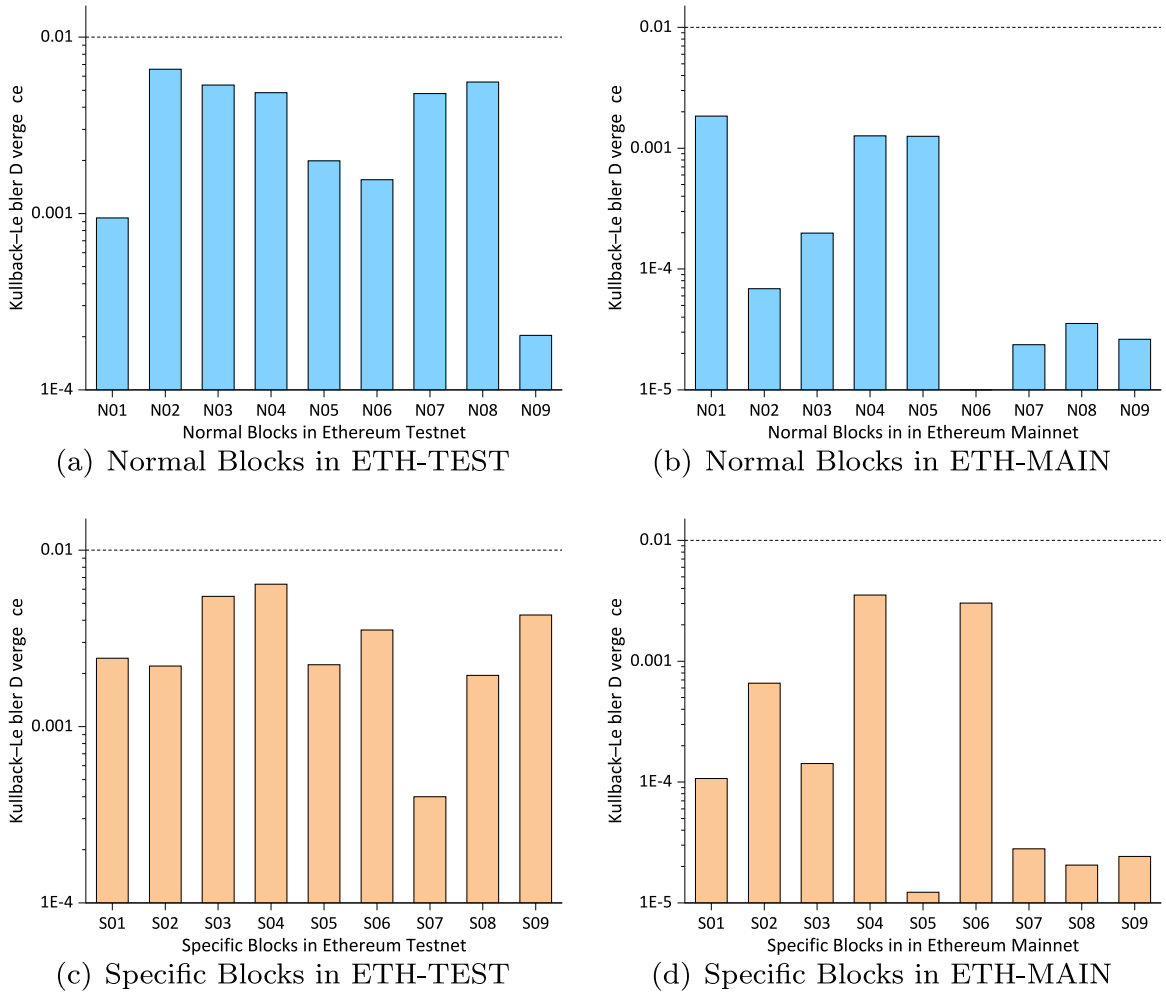


Fig. 8. Result of KLD-Test.

Table 3
Average execution time of each process.

Type	SegNum	Embedding time (ms)	Transmission time (s)	Extraction time (ms)
TestNet	1	0.634	8.5	0.866
	2	1.040	25	2.660
	3	1.516	43	3.210
	4	1.766	44.5	3.312
	5	2.082	36	5.671
MainNet	1	0.792	69	1.154
	2	1.116	132.5	1.858
	3	1.575	65.5	2.821
	4	2.005	353.5	3.834
	5	2.444	102	4.654

the average execution time of each operation as the evaluation results, shown in Table 3.

We can see that as the number of segments increases, the execution time of *Embedding* and *Extraction* processes also increases. The reason is that more segmentation means more specific transaction construction/scanning operations for the sender/receiver during one complete covert communication process. However, the execution time of the above two processes are negligible compared with transmission time. As is shown in Table 3, the transmission time occupies most of the total time of covert communication, which depends on the condition of the blockchain network. TestNet has faster block generation time and lower bandwidth usage compared with MainNet. Hence, it has a faster transmission time.

7.4. Scheme comparison

To further prove the advantages of the proposed scheme, we compare EBDL with existing blockchain-based covert communication schemes from multiple perspectives. The comparison result is shown in Table 4.

We choose eight relevant existing blockchain-based covert communication schemes to make a comparison. More specifically, the schemes in Gao et al. (2020), Cao et al. (2020), Luo et al. (2022) and Tian et al. (2019) are based on the Bitcoin (BTC) blockchain, the schemes in Basuki and Rosiyadi (2019) and Liu et al. (2020) are based on the Ethereum (ETH) blockchain, and the scheme in Liu et al. (2022) is based on the Monero blockchain. As for the scheme in Partala (2018) and the proposed EBDL scheme, they support multiple blockchain platforms such as Bitcoin and Ethereum.

As analyzed above, the most time-consuming process of blockchain-based covert communications is the transmission process. For different schemes, the time cost is not comparable because it mainly depends on the latency of underlying blockchain networks. Hence, we use *Embedding Rate* to make a quantitative comparison of embedding efficiency like exiting works in Basuki and Rosiyadi (2019) and Liu et al. (2020), which is used to make a quantitative comparison between different schemes.

The embedding rate represents the length of the embedded covert message in one transaction, of which the unit is bit/T (embedded bits per transaction). As can be seen in Table 4, EBDL has relative high embedding rate as 512bit/T under the evaluation of *KS-Test* and

Table 4
Scheme comparison.

Scheme	Underlying blockchain platform	Embedding rate (bit/T)	Dynamic label	Large covert message transmission
Gao et al. (2020)	BTC	640	✓	×
Cao et al. (2020)	BTC	1	✓	×
Tian et al. (2019)	BTC	256	✓	×
Luo et al. (2022)	BTC	13	×	×
Basuki and Rosiyadi (2019)	ETH	24	×	×
Liu et al. (2020)	ETH	28	×	×
Liu et al. (2022)	Monero	40	✓	×
Partala (2018)	?	1	×	×
EBDL	?	512	✓	✓

KLD-Test in Section 7.2. In addition, only EBDL supports large covert message transmission because a novel message segmentation mechanism is presented, as shown in Section 5. Although we can further increase the embedded bit-length per transaction, the undetectability will be weaken. Hence, there exists a trade-off between undetectability and embedding efficiency, and we leave this for future work.

Besides, the schemes in Luo et al. (2022), Basuki and Rosiyadi (2019), Liu et al. (2020) and Partala (2018) do not support the dynamic label. More specifically, they use a fixed address or a fixed address set to scan specific transactions, which increases the exposure risk of the sender. The scheme in Gao et al. (2020) leverages the relationship between the last two transactions sent by the same address to scan specific transactions instead of using fixed addresses. However, it sacrifices the transaction scanning efficiency because finding the last two transactions for each address that appeared in a newly-synchronized block is time-consuming.

Based on Table 4 and the above analysis, we can conclude that to the best of our knowledge, the proposed EBDL scheme is the only blockchain-based covert communication scheme that simultaneously supports the dynamic label and large covert message transmission with acceptable efficiency.

8. Conclusion and future work

In recent years, blockchain-based covert communication explores a new research direction with the help of *"no direct connection"* property. However, existing blockchain-based covert communication schemes cannot guarantee realizability, undetectability, or scalability, which means they cannot be used in real-world covert communication scenarios. To solve these issues, we propose EBDL, a blockchain-based covert channel. In EBDL, a novel dynamic label mechanism is presented to scan the specific transactions sent by the sender, which enhances the undetectability because adversaries cannot scan the specific transaction from hundreds of transactions per block. Besides, a segmentation mechanism is presented to support large covert message transmission compared with the existing blockchain-based solutions. Security analysis has proven that EBDL achieves CHA security, and experimental evaluations based on both TestNet and MainNet of Ethereum show that EBDL can resist statistical analysis such as KS and KLD tests with acceptable efficiency.

In future works, we will design more specific transaction construction mechanisms that achieves higher undetectability. For instance, the transaction value field and the signature of the sender can also be used to embed covert messages. Compared with the customized data field, these fields appear in all transactions rather than a portion of them, which further enhance the undetectability. Besides, how to achieve a blockchain-based time covert channel can also be considered an interesting and potential research direction.

CRediT authorship contribution statement

Can Zhang: Conceptualization, Methodology, Software, Writing – original draft. **Liehuang Zhu:** Supervision, Funding acquisition. **Chang Xu:** Resources, Investigation, Formal analysis. **Zijian Zhang:** Project administration, Data curation. **Rongxing Lu:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. U1836212, 61972037, 61402037, 61872041). It is also partially supported by the BIT Research and Innovation Promoting Project (Grant No. 2022YCXZ031).

References

Alsalami, N., Zhang, B., 2018. Uncontrolled Randomness in Blockchains: Covert Bulletin Board for Illicit Activities, Vol. 2018. IACR Cryptol, p. 1184, ePrint Arch..
Association, N.C.S., 1985. Department of Defense Trusted Computer System Evaluation Criteria. Palgrave Macmillan UK.
Basuki, A.I., Rosiyadi, D., 2019. Joint transaction-image steganography for high capacity covert communication. In: 2019 International Conference on Computer, Control, Informatics and Its Applications (IC3INA). pp. 41–46.
Cao, H., Yin, H., Gao, F., Zhang, Z., Khoussainov, B., Xu, S., Zhu, L., 2020. Chain-based covert data embedding schemes in blockchain. IEEE Internet Things J. 1.
Dedic, N., Itkis, G., Reyzin, L., Russell, S., 2009. Upper and lower bounds on black-box steganography. J. Cryptol. 22 (3), 365–394.
Fionov, A., 2019. Exploring covert channels in bitcoin transactions. In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). pp. 0059–0064.
Gao, F., Zhu, L., Gai, K., Zhang, C., Liu, S., 2020. Achieving a covert channel over an open blockchain network. IEEE Netw. 34 (2), 6–13.
Ghosh, A., Gupta, S., Dua, A., Kumar, N., 2020. Security of Cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects. J. Netw. Comput. Appl. 163, 102635.
Gianvecchio, S., Wang, H., 2011. An entropy-based approach to detecting covert timing channels. IEEE Trans. Dependable Sec. Comput. 8 (6), 785–797.
Hopper, N.J., Langford, J., von Ahn, L., 2002. Provably secure steganography. In: Yung, M. (Ed.), Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings, Vol. 2442. In: Lecture Notes in Computer Science, Springer, pp. 77–92.
Iglesias, F., Zseby, T., 2017. Are network covert timing channels statistical anomalies? In: Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, August 29 - September 01, 2017. ACM, pp. 81:1–81:9.
Lampson, B.W., 1973. A note on the confinement problem. Commun. ACM 16 (10), 613–615.

Liang, C., Tan, Y., Zhang, X., Wang, X., Zheng, J., Zhang, Q., 2018a. Building packet length covert channel over mobile VoIP traffics. J. Netw. Comput. Appl. 118, 144–153.

Liang, C., Wang, X., Zhang, X., Zhang, Y., Sharif, K., Tan, Y., 2018b. A payload-dependent packet rearranging covert channel for mobile VoIP traffic. Inform. Sci. 465, 162–173.

Liu, S., Fang, Z., Gao, F., Khoussainov, B., Zhang, Z., Liu, J., Zhu, L., 2020. Whispers on ethereum: Blockchain-based covert data embedding schemes. In: Gai, K., Choo, K.R., Liu, J. (Eds.), BSCI '20: Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure, Co-Located with AsiaCCS 2020, Taipei, Taiwan, October 6, 2020. ACM, pp. 171–179.

Liu, L., Liu, L., Li, B., Zhong, Y., Liao, S., Zhang, L., 2022. MSCCS: A Monero-based security-enhanced covert communication system. Comput. Netw. 205, 108759.

Luo, X., Zhang, P., Zhang, M., Li, H., Cheng, Q., 2022. A novel covert communication method based on bitcoin transaction. IEEE Trans. Ind. Inf. 18 (4), 2830–2839.

Partala, J., 2018. Provably secure covert communication on blockchain. Cryptography 2 (3), 18.

Rowland, C.H., 1997. Covert Channels in the TCP/IP Protocol Suite, Vol. 2. (5), First Monday, URL https://firstmonday.org/ojs/index.php/fm/article/view/528.

Schulz, S., Varadharajan, V., Sadeghi, A., 2014. The silence of the LANs: Efficient leakage resilience for IPsec VPNs. IEEE Trans. Inf. Forensics Secur. 9 (2), 221–232.

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 Netw. Appl. 14 (5), 3029–3042.

Tian, J., Gou, G., Liu, C., Chen, Y., Xiong, G., Li, Z., 2019. DLchain: A covert channel over blockchain based on dynamic labels. In: Zhou, J., Luo, X., Shen, Q., Xu, Z. (Eds.), Information and Communications Security - 21st International Conference, ICICS 2019, Beijing, China, December 15-17, 2019, Revised Selected Papers, Vol. 11999. In: Lecture Notes in Computer Science, Springer, pp. 814–830.

Trabelsi, Z., El-Hajj, W., Hamdy, S., 2008. Implementation of an ICMP-based covert channel for file and message transfer. In: 15th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2008, St. Julien's, Malta, August 31 2008-September 3, 2008. IEEE, pp. 894–897.

Tschorsch, F., Scheuermann, B., 2016. Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. 18 (3), 2084–2123.

Wang, Y., Chen, P., Ge, Y., Mao, B., Xie, L., 2009. Traffic controller: A practical approach to block network covert timing channel. In: Proceedings of the the Forth International Conference on Availability, Reliability and Security, ARES 2009, March 16-19, 2009, Fukuoka, Japan. IEEE Computer Society, pp. 349–354.

Wang, Z., Huang, L., Yang, W., He, Z., 2017. A classifier method for detection of covert channels over LTE. In: 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2017, Nanjing, China, October 12-14, 2017. IEEE, pp. 454–460.

Zhang, X., Liang, C., Zhang, Q., Li, Y., Zheng, J., Tan, Y., 2018. Building covert timing channels by packet rearrangement over mobile networks. Inform. Sci. 445–446, 66–78.

Zhang, L., Zhang, Z., Jin, Z., Su, Y., Wang, Z., 2021. An approach of covert communication based on the Ethereum whisper protocol in blockchain. Int. J. Intell. Syst. 36 (2), 962–996.

Zhang, L., Zhang, Z., Wang, W., Jin, Z., Su, Y., Chen, H., 2022. Research on a covert communication model realized by using smart contracts in blockchain environment. IEEE Syst. J. 16 (2), 2822–2833.

Zhang, Q., Zhang, X., Xue, Y., Hu, J., 2020. A stealthy covert storage channel for asymmetric surveillance VoLTE endpoints. Future Gener. Comput. Syst. 102, 472–480.

Zhang, X., Zhu, L., Wang, X., Zhang, C., Zhu, H., Tan, Y., 2019. A packet-reordering covert channel over VoLTE voice and video traffics. J. Netw. Comput. Appl. 126, 29–38.

Can Zhang received his B.E. (Bachelor of Engineering) degree in Computer Science and Technology from Beijing Institute of Technology, Beijing, China, in 2017. He is currently a Ph.D. student at the School of Cyberspace Science and Technology, Beijing Institute of Technology. His current research interests include security & privacy in VANET, cloud computing security, and blockchain technology.

Liehuang Zhu received his Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004. He is currently a professor at the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, cloud computing, and blockchain applications.

Chang Xu received her Ph.D. degree in computer science from Beihang University, Beijing, China, in 2013. She is currently an associate professor at the School of Cyberspace Science and Technology, Beijing Institute of Technology. Her research interests include security & privacy in VANET, and big data security.

Zijian Zhang received the Ph.D. degree from the School of Computer Science and Technology, Beijing Institute of Technology. He is currently a Research Fellow of the School of Computer Science, The University of Auckland. He is also with the School of Cyberspace Science and Technology, Beijing Institute of Technology. He was a Visiting Scholar with the Computer Science and Engineering Department, State University of New York at Buffalo in 2015. His research interests include design of authentication and key agreement protocol and analysis of entity behavior and preference.

Rongxing Lu is an associate professor at the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. Dr. Lu is an IEEE Fellow. His research interests include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy. He has published extensively in his areas of expertise (with H-index 72 from Google Scholar as of November 2020), and was the recipient of 9 best (student) paper awards from some reputable journals and conferences. Currently, Dr. Lu serves as the Vice-Chair (Conferences) of IEEE ComSoc CIS-TC (Communications and Information Security Technical Committee).