mathematics

Article

A Blockchain-Enabled Group Covert Channel against Transaction Forgery

Tongzhou Shen [1], Liehuang Zhu [1], Feng Gao [1], Zhuo Chen [1], Zijian Zhang [1,]* and Meng Li [2,]*

[1] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China; 3120211366@bit.edu.cn (T.S.); liehuangz@bit.edu.cn (L.Z.); chen.zhuo@bit.edu.cn (Z.C.)

[2] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China

* Correspondence: zhangzijian@bit.edu.cn (Z.Z.); mengli@hfut.edu.cn (M.L.)

Abstract: As a decentralized network infrastructure, the data sent to the blockchain are public and temper-evident. The cover of massive normal transactions in a blockchain network is ideal for constructing a stable and reliable covert channel to achieve one-to-many group covert communication. Existing blockchain-based covert communication schemes face challenges in balancing concealment, embedding rate and filtering efficiency, making them unsuitable for direct extension to group scenarios. Adopting a key-leakage scheme can increase the channel capacity while maintaining high concealment from external adversaries. However, it will also expose more knowledge to the receiver. A malicious receiver has the ability to steal a sender's identity or replay historical transactions to control the entire channel. In this paper, we define the capabilities of malicious receivers in blockchain-based group covert communication scenarios and propose a group covert communication scheme resistant to transaction forgery attacks. Theoretical analysis and experiments prove that our covert transactions do not have any transaction correlativity, ensuring the unique authenticity of the sender's identity while maintaining supreme concealment compared with the existing schemes. The precision and recall of machine learning detection results can reach 0.57–0.62 (0.5 is the ideal value).

Keywords: blockchain network; steganography; covert communication; covert channel

MSC: 68P27

Citation: Shen, T.; Zhu, L.; Gao, F.; Chen, Z.; Zhang, Z.; Li, M. A Blockchain-Enabled Group Covert Channel against Transaction Forgery. *Mathematics* **2024**, *12*, 251. https://doi.org/10.3390/math12020251

Academic Editor: Daniel-Ioan Curiac

Received: 13 December 2023
Revised: 7 January 2024
Accepted: 8 January 2024
Published: 12 January 2024

1. Introduction

Blockchain-based covert communication utilizes the blockchain transactions and network as a covert channel for steganographic transmission [1]. Specifically, the sender builds covert transaction by embedding the message within certain fields of a typical blockchain transaction. Meanwhile, the receiver needs to identify the covert transaction from all of the transactions recorded on the blockchain and subsequently extract the encoded message. Traditional network-based covert communication mechanisms rely on peer-to-peer transmission, which can expose the identities of both communicating parties. Covert channels established through the utilization of packet redundancy and inter-packet delays exhibit instability. They are prone to detection via data statistical analysis and machine learning techniques. Moreover, they can be disrupted by adversarial maneuvers such as packet reordering and noise injection attacks [2]. Hence, their security and stability cannot be adequately guaranteed. In contrast, the blockchain provides a decentralized and distributed system that leverages the immutability of transaction and the randomness of address, which brings high confidentiality and reliability to the communication process.

In a one-on-one blockchain-based covert communication scenario, the existing systems focus on security against external adversaries. These adversaries mainly use data statistical analysis methods to identify covert transactions in the blockchain, such as using KLD (Kullback–Leibler divergence) tests and KS tests to detect whether the character distribution is abnormal. To counter these detection methods, the sender collects the distribution patterns of normal transactions and constructs covert transactions that match them, or they embed covert messages using fields with unordered outputs, such as signatures [3].

In addition, since blockchain transaction data are public, anyone can check historical transaction data. Therefore, receivers can filter covert transactions offline instead of performing operations on the accounts. The sender can negotiate the necessary parameters for communication with any number of receivers and then achieve group covert communication between the sender and multiple receivers through the blockchain by making a single covert transaction. All receivers can filter out covert transactions at any time since they are recorded.

In general, the sender assigns different key pairs for each receiver group as identity [4]. When communicating with a certain receiver group, the sender chooses the group's corresponding key pair to participate in generating the covert transaction. This mechanism ensures that only members of the specified group can receive and extract messages. Since the selected algorithm remains the same each time, the construction pattern of the covert transaction is fixed.

In a blockchain-based group covert communication scenario, where multiple receivers or receiver groups exist, the threat of internal adversaries needs to be considered. These internal adversaries are more interested in messages sent to other receiver groups and attempt to use knowledge on the receiver side to reconstruct the entire covert channel. They may even collude with external adversaries, resulting in more severe security threats to covert communication. Since they do not possess the corresponding keys, internal adversaries cannot accurately identify covert transactions using filtering algorithms. However, relative to external adversaries, internal adversaries are familiar with the characteristics of covert transactions generated by the same algorithm, such as the number of inputs and outputs in a transaction and the way addresses are associated. As a result, they are more likely to identify suspected covert transactions sent to other groups. Based on the above analysis, the sender needs to not only simulate normal transactions on character distribution but also strengthen the covert nature of transaction patterns, minimize the correlativity between transactions, and reduce the security risks posed by internal adversaries.

On the other hand, to achieve efficient embedding and filtering, the sender may choose a key-leakage-based scheme to implement covert communication, which exposes the sender's private key on the receiver's side. In this case, malicious receivers may use the sender's private key to send covert transactions and steal the sender's identity to communicate with other receiver groups or replay historical transactions in the channel to carry out an identity-forgery attack and ultimately control the entire channel. Thus, in a blockchain-based group covert communication scenario, an effective authentication mechanism is needed to ensure the unforgeability of the sender's identity and transactions.

A well-designed blockchain-based covert communication scheme should be able to resist attacks from internal adversaries while maintaining high channel capacity and efficiency without sacrificing anonymity. Existing covert communication models underestimate the threat posed by internal adversaries and lack effective measures to counter identity forgery attacks. In addition, existing covert communication schemes exhibit significant differences in transaction association and patterns from ordinary transactions, which can be easily exploited by adversaries to expose the covert channel. To address these issues, we propose a novel identity authentication mechanism based on the BLS short signature [5]. The sender generates an identity authentication signature, and the receiver can uniquely determine the source of the message based on the validity and timeliness of the signature, thereby excluding interference from internal adversaries. Moreover, to reduce the correlativity of transactions while efficiently applying the mechanism, we propose an message-embedding and transaction filtering algorithm based on Kleptography and ECDH (Elliptic Curve Diffie–Hellman). By integrating the two and embedding the identity authentication signature in the transaction, we ultimately design a blockchain-based group covert communication scheme that is effective against attacks from internal adversaries.

We evaluate the scheme's concealment from the perspectives of defending against both external and internal adversaries. The results of KLD and KS tests show that the covert transactions generated by the scheme can resist statistical analysis attacks. An internal adversary may obtain extra accuracy in locating covert transactions sent to other groups with known transaction features. We use a machine learning detection model to simulate the attack. The results of the machine learning detection show that the internal adversary cannot effectively distinguish between covert and ordinary transactions even if they obtain transaction construction patterns.

Our contributions:

- We propose a blockchain-based group covert communication model, which takes into account the threat from internal adversaries to lay the foundation for solving the transaction detection and identity forgery risks.
- We propose an message embedding and transaction filtering mechanism which can achieve data covert embedding and extraction while ensuring the unlinkability of transactions, thereby enhancing the concealment of covert transactions. An identity authentication mechanism based on the BLS signature is also implemented within the limited storage space of blockchain transactions to achieve sender identity authentication and address the risks of identity forgery and replay attacks by internal adversaries.
- We implemented a prototype system and evaluated the proposed scheme to demonstrate the feasibility. Experimental results show that compared to the existing method, our scheme can resist identity forgery attacks by internal adversaries, achieve higher concealment against external adversaries and provide a higher channel capacity with acceptable cost.

The remainder of this paper is organized as outlined below. Section 2 proposes our blockchain-based group covert communication system and threat model. Section 3 provides related works about traditional covert communication, blockchain-based covert communication and blockchain-based group covert communication. Section 4 discusses the impact of transaction correlation on the concealment and introduces relevant background knowledge. Section 5 describes the implementation details of the proposed scheme. In Section 6, we evaluate the performance of our scheme and analyze the countermeasures against the adversaries. Finally, Section 7 concludes the paper and proposes future research directions.

2. Problem Statement

Traditional network-based covert channels are susceptible to detection and disruption, while blockchain-based covert communication offers inherent concealment and anonymity. In current blockchain-based group covert communication models, the attack target of internal adversaries is defined as discovering covert channels, but the possibility of internal adversaries forging covert transactions is overlooked. For example, the sender sends covert transactions from a fixed address [6], which is the sole identifier used by the receiver to filter transactions. However, the private key corresponding to this address can be derived by the receiver, so once the algorithm is public, all receivers can create covert transactions and send them from that address. To ensure the unique authenticity of the sender's identity in the context of group covert communication, this section presents a blockchain-based group covert communication model that can resist forged transaction attacks. In another scheme, the sender uses a group key to create the corresponding index matrix of address interaction [4]. The receiver uses the same group key to recover the matrix and filter transactions. As long as malicious receivers have knowledge of the transaction construction algorithm, they have the capability to forge covert transactions using the held group key, thus deceiving other members of the group. Related works are detailed in Section 3.

In this section, we present the system model as shown in Figure 1 alongside a formal description of the threat model and the design goals of our scheme.

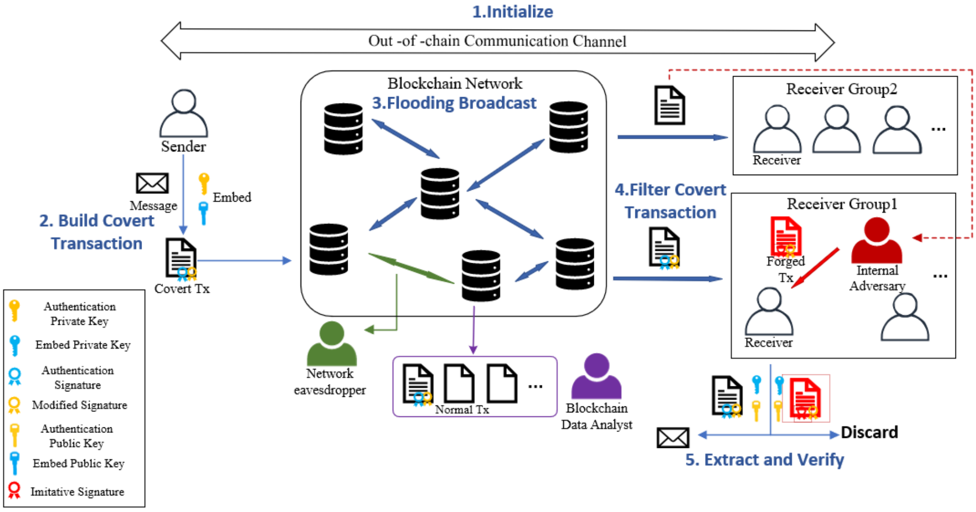


Figure 1. System model.

2.1. System Model

A complete blockchain-based group covert communication process consists of five steps including initialize, build covert transaction, flooding broadcast, filter covert transaction, extract and verify. A one-time out-of-chain communication channel and three entities participate in the process: the sender, the receiver, and the blockchain network. The entities of the system model are given below:

- Sender: The sender generates the necessary parameters and initiates group covert communication on the blockchain. During a communication session, the sender embeds covert messages and identity authentication signatures into a blockchain transaction, transforming it into a covert transaction. Then, the sender sends this transaction from a blockchain node to the network, thereby transmitting the covert messages.
- Receiver: The receiver filters covert transactions from the blockchain based on pre-negotiated parameters, verifies their validity, and extracts the covert messages. In our proposed system model, receivers exist in the form of groups. All receivers within the same group hold the same parameters, while the process of filtering transactions remains independent. The sender generates transactions using the key corresponding to the designated receiver group to achieve covert communication.
- Out-of-chain communication channel: The secure channel outside the blockchain that the sender and receiver utilize before establishing their initial communication. Through this channel, the sender transmits the necessary parameters for establishing communication such as keys, shared random numbers, and so on. This channel does not involve any actual communication content. For security, it is only used once unless the sender needs to change the key to construct a new channel.
- Blockchain network: The media used for covert communication. Covert transactions originate from one blockchain node and propagate to all nodes in the blockchain network through the flooding broadcast mechanism. Once these transactions are confirmed, they will be recorded in the transaction list of a particular block.
- Initialize: Prior to the first communication, the sender assigns a group label to the receiver and negotiates necessary parameters.
- Build covert transaction: The sender encodes the original information and constructs a covert transaction using an embedding algorithm. This covert transaction contains the sender's identity authentication signature, proving that it originates from the sender. Additionally, the transaction includes a modified signature to assist the receiver in filtering and extracting covert information.
- Flooding broadcast: The covert transaction is broadcasted throughout the network using a flooding method and is eventually recorded on the blockchain.

- Filter covert transaction: The receiver searches the public blockchain for covert transactions based on the modified signature and the receiver's group label.
- Extract and verify: The receiver extracts the original information from the covert transaction and verifies the sender's identity authentication signature. If the signature is valid, it confirms that the transaction is from the legitimate sender. Otherwise, if the signature is invalid, it indicates that the transaction is a forged transaction with an imitative signature created by an internal adversary.

2.2. Threat Model

In our blockchain-based group covert communication system, adversaries that need to be considered can be summarized into three main types, including network eavesdropper, blockchain data analyst and internal adversary. Network eavesdropper and blockchain data analyst are classified as external adversaries.

- Network eavesdropper: Adversaries present in the form of blockchain network nodes that propagate transactions. They monitor the network flow on its connected nodes. These adversaries attempt to identify covert channels, recognize the communication parties and extract covert messages by intercepting and analyzing network traffic packets, even reconstructing the entire communication network topology.
- Blockchain data analyst: Adversaries analyze transaction data to summarize characteristics from all blocks. Their goal is to differentiate between normal transactions and covert transactions. When more covert transactions are uncovered, analysts are increasingly able to identify the properties of covert channels, thereby detecting communication behaviors.
- Internal adversary: Internal adversaries present as malicious receivers within the receiver group. As communication participants, internal adversaries possess the same parameters and permissions as legitimate receivers. Once they obtain enough knowledge like a sender's private key, they can send forged covert transactions by imitating the construction pattern of the sender. Specifically, they build covert transactions with modified signatures and imitative signatures, which can be recognized by other receivers. This allows them to communicate with other receivers in the same group, thereby delivering incorrect information. Alternatively, they can replay transactions previously sent by the sender at any given time, therefore destroy message timeliness. The receivers cannot distinguish the authenticity of the message source due to the attacks which make the channel untrustworthy. Furthermore, compared to external adversaries, internal adversaries are quite familiar with the characteristics of cover transactions. This facilitates their ability to identify and target covert transactions sent to other groups.

2.3. Design Goals

Based on the formal models and definitions given above, we introduce the design goals of the proposed scheme as follows:

- Concealment: Covert transactions should exhibit indistinguishability, where internal adversaries are unable to discern transactions intended for other groups, and external adversaries cannot differentiate between covert and regular transactions.
- Practicability: Covert transactions should be efficiently sent and filtered. The time required for covert transaction filtration should be less than the time it takes to generate a new block. The time costs associated with information embedding and extraction should be acceptable.
- Unforgeability: In covert communication, the identity of the sender should be uniquely verifiable, and internal adversaries should be incapable of forging sender transactions using recipient parameters.
- Capability: Covert communication should provide a sufficient channel capacity while ensuring concealment, practicability and unforgeability. The channel should facilitate the cost-effective transmission of covert messages.

3. Related Work

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

3.1. Traditional Covert Communication

Covert communication serves as a mechanism for discreet information exchange between senders and receivers via public channels, eluding detection by third parties [7]. Traditional covert communication methods often rely on the establishment of covert channels using network traffic data packets or time intervals. In covert storage channels, various fields within network protocols, such as the ACK or SEQ fields of the TCP/IP protocol, are exploited for storing covert information [8]. Covert time channels leverage the order of data packet arrivals or the intervals between them for transmitting covert information [9]. Unfortunately, these fields exhibit fixed positions, rendering them susceptible to detection, restriction, or even modification by potential attackers, thereby compromising the integrity of the communication [2]. Furthermore, this approach necessitates the establishment of prolonged direct communication between the involved parties and is significantly impacted by fluctuations in network latency, rendering it inherently unreliable.

Traditional network-based covert channels often necessitate the establishment of direct communication between communicating parties, where their IP addresses are embedded within network data packets. Consequently, this direct association compromises the stealthiness of the communication, and the susceptibility of the communication carrier to interception and tampering further diminishes the reliability of traditional covert communication methods.

3.2. Blockchain-Based Covert Communication

Blockchain-based covert communication utilizes blockchain transactions as carriers of concealed information, achieving information dissemination through the propagation of transactions within the blockchain network. Blockchain transactions employ a flooding broadcast mechanism for network transmission, which is characterized by non-directional dissemination. Transactions embedded with covert information undergo forwarding by multiple nodes, eventually broadcasting across the entire network, making it challenging to ascertain the originating IP address of the transaction. Consequently, direct communication between the sender and receiver is not established. The immutability of blockchain transactions is ensured through the application of hash algorithms and consensus mechanisms. Furthermore, the blockchain addresses possess anonymity. In comparison to traditional covert channels, blockchain-based covert channels inherently exhibit concealment and reliability.

Zombiecoin [10] is an architecture for botnets implemented in the Bitcoin main network. The sender employs a fixed address to send covert transactions and embeds covert messages in the custom field OP_RETURN. With the advancement of abnormal transaction detection techniques in the blockchain, the intended padding of OP_RETURN fields reduces the concealment [11], and the reuse of addresses becomes detectable through address clustering methods [12]. Chainchannel [13] is a covert channel based on secret sharing and private key leakage. In each communication, the sender needs to send several transactions with the same address to completely leak covert messages. After each communication, additional transactions must be sent from the fixed address to update the necessary parameters for the next communication, leading to address reuse and a strong correlativity between covert transactions. Fionov [14] encoded the message as a random number used for signing and embedded it into the digital signature generated by the ECDSA algorithm. The sender needs to inform the receiver in advance of the private key corresponding to the input address of the covert transaction. This method has poor robustness. Once exposed, the covert channel can not be replaced in time. Tian et al. [15] proposed a covert communication scheme called DLchain based on a dynamic label algorithm. In this scheme, the sender and receiver dynamically generate synchronized OP_RETURN field content based on the statistical distribution of current blockchain transactions in OP_RETURN fields. However, the sender uses the same random number to sign two covert transactions, resulting in a partially identical signature, making them distinguishable from regular transactions. Cao et al. [16] proposed a covert message-embedding scheme based on a chained relationship between input addresses and output addresses. The output address of a covert transaction serves as the input address for the next covert transaction. The sender is able to embed 1 bit of data in each transaction. However, this method lacks scalability, as increasing the embedding rate comes at a computational cost that the sender cannot afford. In covert communication based on the Whisper protocol [17], the topic field is used as the unique identifier for the receiver to filter transactions. However, if the receiver fails to filter the transaction before it expires, the covert message may be lost.

From the above analysis, while blockchain-based covert channels can prevent the tampering of concealed information, they still face challenges such as address reuse, explicit embedding, and low channel capacity.

3.3. Blockchain-Based Group Covert Communication

Blockchain-based covert communication involves two critical steps: the embedding of covert messages and the filtration of covert transactions. Since transactions are publicly accessible, the filtration process of covert transactions is executed off-chain. Consequently, covert transactions can be filtered by any number of receivers. Thus, in scenarios involving multiple receivers or receiver groups, blockchain-based group covert communication can be realized. Typically, the sender employs distinct keys to identify group identities and communicates with all members within the group [18].

Generally, owing to different group keys, receivers can only filter transactions intended for their respective groups. Nonetheless, malicious receivers, as internal adversaries, attempt to steal the sender’s identity, forge transactions purportedly sent to their own group, or identify transactions sent to other groups. Gao et al. [6] combined kleptography techniques with ECDSA signatures in Bitcoin to implement a mechanism that can filter indistinguishable transactions in polynomial time. However, the sender generates all covert transactions from a fixed address. Furthermore, all the parameters required for creating a covert transaction can be inferred by the receiver from known parameters. For example, while filtering transactions, the recipient can calculate the private key corresponding to the fixed address. A malicious receiver is fully capable of controlling the funds in that address and sending out their own constructed transactions, thereby forging the identity of the sender and engaging in covert communication with other receivers. Zhang et al. [4] proposed a group covert communication scheme based on the index matrix of address interaction. However, to ensure concealment, this scheme imposes limitations on the number of transactions sent within a single block.

The majority of existing blockchain-based covert communication schemes can be extended to group scenarios. However, in addition to the inherent challenges related to concealment, correlativity and channel capacity, these schemes lack effective measures to resist threats from internal adversaries. Therefore, this paper proposes a scheme that can withstand internal adversary threats while ensuring high concealment and channel capacity. This scheme proves to be effective in the context of covert communication within group scenarios.

4. Preliminaries

In this section, we introduce the relevant background knowledge of our scheme.

4.1. Transaction Correlation

In this subsection, we discuss the transaction correlation and describe its impact on concealment.

4.1.1. Overview

In Bitcoin, the transfer of funds follows the Unspent Transaction Output (UTXO) model. The input UTXO typically includes an input address, the sender's signature, and the transferred amount, while the output UTXO commonly includes an output address or public key and the transferred amount. Notably, UTXOs containing the OP_RETURN field can be classified as null-data type and exhibit prominent features.

Bitcoin recommends the use of disposable addresses, which contributes to a significant portion of transactions with one input UTXO and two output UTXO [19]. Consequently, this model inherently provides concealment when constructing covert transactions. However, all funding sources must be transferred from one or multiple previous transactions' UTXOs. As a result, transaction traceability can be established by examining the addresses and corresponding amounts spent in the UTXOs, ultimately forming a transaction chain.

Transactions carrying covert messages exhibit a notable characteristic: the output addresses often include at least one controlled by the sender, serving as the change address. In this way, the sender can accomplish covert communication by solely expending the transaction fee. The UTXO transferred to controllable output addresses will subsequently serve as input UTXO for future covert communication. Therefore, if an adversary identifies any of these transactions, they can easily discover additional covert transactions by examining the associated addresses and the chain relationship of the transactions. This facilitates further verification and analysis toward covert channels.

Assuming that the sender and receiver establish the essential parameters through a single negotiation prior to conducting on-chain communication, all covert communication is solely based on this initial agreement. Additionally, the sender possesses an abundant number of Bitcoin addresses with ample funds, disregarding any financial links unrelated to covert communication. The interrelation between covert transactions can be categorized into static-address correlation and chain-address correlation.

4.1.2. Static-Address Correlation

If an address $addr_i$ is involved in multiple covert transactions, these covert transactions exhibit static-address correlation. As shown in Figure 2, assuming an adversary discovers a specific covert transaction TX_i, the adversary can easily retrieve related transactions (TX_1, TX_2, TX_3) by filtering $addr_i$. In this way, the adversary can obtain all covert transaction samples. Since the reuse of addresses violates the disposable address rule in Bitcoin, this characteristic significantly reduces the concealment of the covert channel. Static-address correlation is mainly caused by the vulnerable system design. For instance, a receiver may use a fixed address as a label to filter covert transactions [10,14]. The sender may also need to use the same address to send transactions in a single communication process, thereby exposing the private key [6,15] or improving insufficient channel capacity [13].

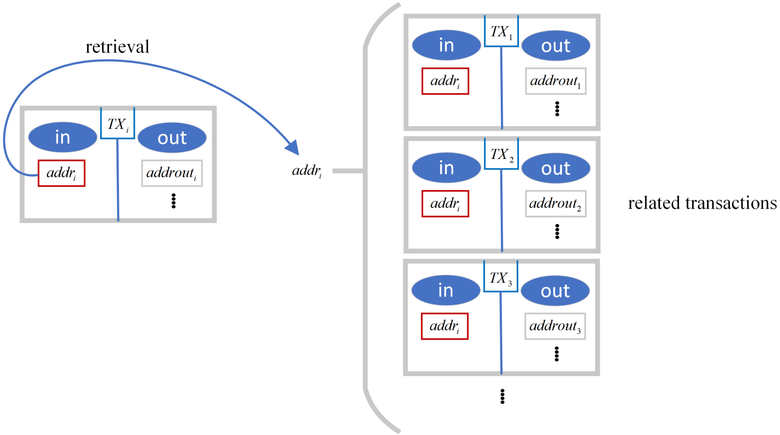


Figure 2. Static-address correlation.

4.1.3. Chain-Address Correlation

Sending transactions from dynamic addresses can avoid static-address correlation, but it also sacrifices efficient filtering ability. In order to ensure that the receiver can still recognize covert transactions, existing schemes utilize key or address derivation relationships to generate covert transactions [16]. As shown in Figure 3, TX_j is one of the transactions in the set of covert transactions TX. Due to the derived relationship between the keys used to construct the covert channel, the addresses in covert transactions automatically form an indexing sequence. The output address of the current covert transaction will be used as the input address for the next covert transaction; otherwise, the receiver will not be able to recognize the covert transaction or correctly extract the covert messages. In this case, all covert transactions can be connected as a chain of transactions based on the sequential relationship between the addresses. An adversary can start from TXj, find the previous covert transaction TX_{j-1} based on the input address $addr_j$ and then trace all historical transactions. Alternatively, the adversary can start from the output address $addr_{j+1}$, monitor and then find transactions like TX_{j+1}, TX_{j+2} that are sent after TX_j. Therefore, the adversary will find it easier to monitor the channel. In general, chain-address correlation is more likely to exist between the transactions sent in successive communication.

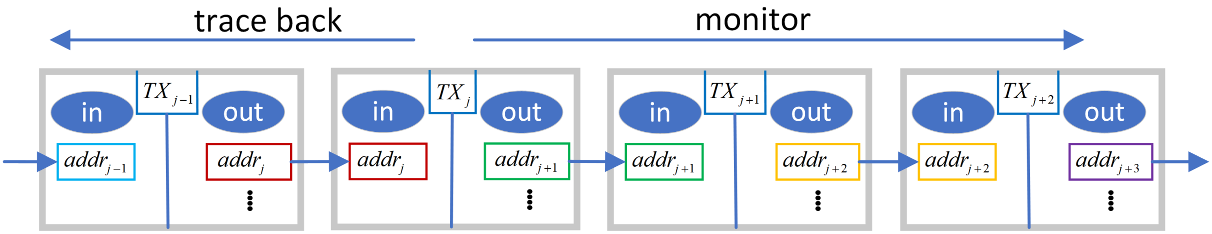


Figure 3. Chain-address correlation.

4.1.4. Unlinkability of Transactions

To achieve the unlinkability of transactions, two objectives need to be met: the absence of static-address correlation within a single or successive communication and the absence of chain-address correlation across historical communications. As shown in Figure 4, NTX represents a set of normal transactions, while CTX represents a set of covert transactions. They are distributed along four transaction chains depicted by the relationships between addresses and UTXO inputs and outputs. In general, to achieve one covert communication, it may be necessary to send more than two transactions. In the context of a single communication, the covert transactions involved should not exhibit static-address correlation. For example, CTX_{p_1} and CTX_{p_2} represent two transactions sent during the p-th communication, and the adversary should not be able to link them based on input addresses. Additionally, for each group of covert transactions involved across different communications, such as the p-th group (CTX_{p_1}, CTX_{p_2}) and the q-th group (CTX_{q_1}, CTX_{q_2}), they should not exhibit chain-address correlation. In other words, the adversary should not be able to find any history or traces of the covert channel based on the covert transactions sent during each communication. To satisfy these requirements, it can be concluded that each covert transaction should be distributed on an independent transaction chain. Furthermore, within each covert transaction and its corresponding transaction chain, duplicate addresses should not be used in the inputs and outputs. This way, the adversary will not be able to find any other covert transactions based on a single covert transaction through the transaction chain.

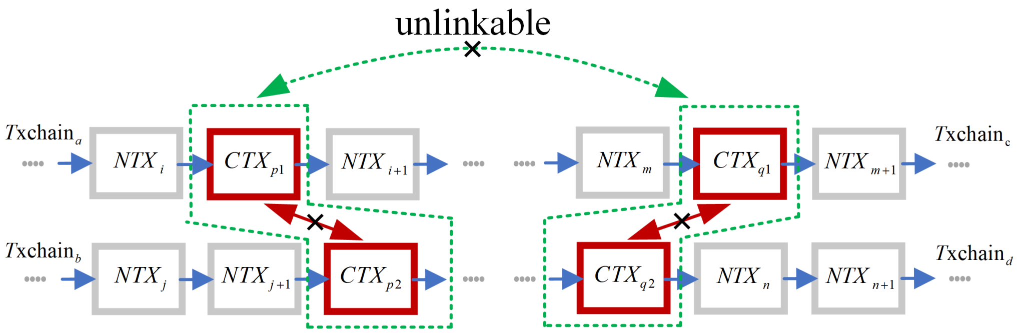


Figure 4. Unlinkability of transactions.

4.2. ECDSA

The proposed blockchain group covert communication scheme utilizes the ECDSA (Elliptic Curve Digital Signature Algorithm) signature in Bitcoin for implementation. In this section, we provide an explanation of the algorithm's process. Subsequently, in the specific description of the scheme, we only discuss the operations involved with the algorithm's parameters and the direct acquisition of its input and output. It should be noted that this scheme can be implemented not only in Bitcoin but also in any blockchain that utilizes the ECDSA algorithm for digital signatures, such as Ethereum. Bitcoin uses the secp256k1 elliptic curve to implement the ECDSA signature algorithm. The globally shared domain parameters include the generator point G and the order of the curve N. The elliptic curve multiplication is represented as $(\cdot)$ and the transaction hash value to be signed is described as h.

We compute the ECDSA signature $(r, s) = sign_{ECDSA}(d, k_1, h)$ by the following steps:

(1) Choose a random integer $d \in [1, N]$ to calculate the point Q on the elliptic curve; d is the private key and Q is the public key.

$$Q = d \cdot G \tag{1}$$

(2) Choose a random integer $k \in [1, N]$ to calculate the point P on the elliptic curve.

$$P = (r, y) = k \cdot G \tag{2}$$

(3) Compute

$$s = k^{-1} * (h + d * r) \bmod N \tag{3}$$

In Equation (3), only d and k cannot be directly obtained or computed from transaction data. In other words, if one of them is known, any one can deduce the other parameter. Therefore, if embedding covert messages in d or k, the key lies in ensuring that only the receiver can access them.

4.3. Kleptography on ECDSA

Kleptography is a technique that embeds messages and modifies the original cryptographic system algorithm with the attacker's key to produce output that is indistinguishable from the original output, thus enabling the secure and subliminal stealing of information [20]. Existing schemes have implemented covert channels in Bitcoin using kleptography [6]. In these schemes, the sender generates a private key and calculates $pk_s = sk_s \cdot G$ on the elliptic curve, which are then sent to the receiver through a one-time secure out-of-chain channel. Subsequently, the sender modifies the ECDSA algorithm in Bitcoin to generate a special signature that is indistinguishable from the normal one while ensuring the validity. The modified signature algorithm $sign_{ECDSA-KLE}(d, h_1, h_2, pk_{kle})$ and verification algorithm $extract_{ECDSA-KLE}(\sigma_1, \sigma_2, h_2, sk_{kle})$ are shown below.

As shown in Algorithm 1, the sender chooses a private key d to create two unsigned transactions tx_1 and tx_2; then, it calculates their corresponding unsigned transaction hash value h_1 and h_2. A random binary string k_1 with length λ is chosen as a random factor, which is 256 bits in Bitcoin. The first transaction tx_1 is signed by using normal ECDSA. Next, a deterministic function map is used with inputs k_1 and h_2 to obtain the random factor k_2, which is required for signing the second transaction tx_2. This algorithm generates two signatures at one time.

Algorithm 1 $sign_{ECDSA-KLE}(d, h_1, h_2, pk_{kle})$

```
Require: $d, h_1, h_2, pk_{kle}$
Ensure: $(\sigma_1, \sigma_2)$
Randomly choose $k_1 \leftarrow \{0,1\}^{\lambda}$
Set $\sigma_1 = (r_1, s_1) = sign_{ECDSA}(d, k_1, h_1)$
Set $k_2 = map(k_1, pk_{kle}, h_2)$
Set $\sigma_2 = (r_2, s_2) = sign_{ECDSA}(d, k_2, h_2)$
return $(\sigma_1, \sigma_2)$
```

The receiver identifies covert transactions by executing Algorithm 2. Firstly, the receiver checks the value r_1 from the signature of the first transaction tx_1 as the x-coordinate to find the point X on the elliptic curve. This point can also be obtained during the process of verifying the validity of the ECDSA signature [21]. Secondly, the receiver uses the deterministic function $map^{'}$ with inputs X and h_2 to derive the random number k_2 chose by the sender to sign the second transaction tx_2. Finally, the receiver can compute the private key d used to send these two transactions by Equation (3). In practical scenarios, the receiver can select any two transactions from the blockchain, assuming that they are the two covert transactions generated by the sender and apply the extraction algorithm to attempt to obtain private key d. The receiver uses d to generate a Bitcoin address, and if that address matches the one exactly used to send the transactions, it confirms that these two transactions carry covert messages. Otherwise, they need to continue searching for a pair of transactions that meet the criteria.

Algorithm 2 $extract_{ECDSA-KLE}(\sigma_1, \sigma_2, h_2, sk_{kle})$

```
Require: $\sigma_1, \sigma_2, h_2, sk_{kle}$
Ensure: $d$
draw $X = (r_1, y_1)$ from $\sigma_1$
Set $k_2 = map^{'}(X, sk_{kle}, h_2)$
Set $d = r^{-1} * (s_2 * k_2 - h_2) \bmod N$
return $d$
```

Assuming there are n transactions in a block, the receiver can use the aforementioned method to filter the covert transactions (tx_1, tx_2) with a time complexity of $O(n^2)$. To reduce computation and improve the filtering efficiency, the receiver can first collect a set of transactions $TX = \{Tx_1, Tx_2, \ldots\}$ related to the negotiated fixed address from the blockchain and perform the filtering operation by iterating through the set. However, using a fixed address significantly decreases the concealment of the covert transactions [12]. Furthermore, utilizing a fixed address implies that no information can be embedded within the private keys of the two transactions, requiring extra alternative user-controlled fields, such as OP_RETURN and other custom fields, to achieve information embedding, which further reduces concealment [11].

Constructing covert channels within the blockchain poses compatibility challenges in terms of concealment, channel capacity, and filtering efficiency. We draw upon kleptography techniques and introduce enhancements to implement a covert communication scheme with variable sender addresses. The scheme supports implicit embedding and ensures that covert transactions cannot be differentiated from regular transactions. We also reduce the filtering efficiency to $O(n)$.

4.4. Ecdh in Covert Communication

In the ECDHC-CDE scheme [16], the Elliptic Curve Diffie–Hellman (ECDH) algorithm is utilized to establish the derivation relationship between private keys and public keys. Specifically, the new derived private key has to be used for generating output addresses in the same transaction. In fact, this concept can effectively establish secret connections between any two transactions by employing a pre-shared key (PSK) denoted as k_s. In blockchain-based group covert communication, k_s refers to the key and label that the sender distributes to the receiver group. Different receiver groups correspond to different k_s values. The sender builds transactions using different k_s values to communicate with the respective groups. Only the groups chosen by the sender can link two covert transactions, allowing for the filtration of covert transactions.

$$sk_2 = sk_1 * k_s \bmod N \tag{4}$$

$$pk_1 = sk_1 \cdot G \tag{5}$$

$$pk_2 = sk_2 \cdot G \tag{6}$$

$$pk_2 = pk_1 \cdot k_s = (sk_1 \cdot G) \cdot k_s = (sk_1 * k_s) \cdot G = sk_2 \cdot G = pk_2 \tag{7}$$

As shown in Figure 5, the sender randomly generates private key sk_1 for transaction TX_1. Subsequently, the private key sk_2 for transaction TX_2 is computed as Equation (4). The corresponding pk_1 and pk_2 can be derived from Equations (5) and (6). Equation (7) derives the correctness of obtaining pk_2 using pk_1 and k_s. Since pk_1 is public on the blockchain, once retrieving the first transaction TX_1 from the blockchain, the receiver can calculate pk_2 by Equation (7), thereby filtering out TX_2.

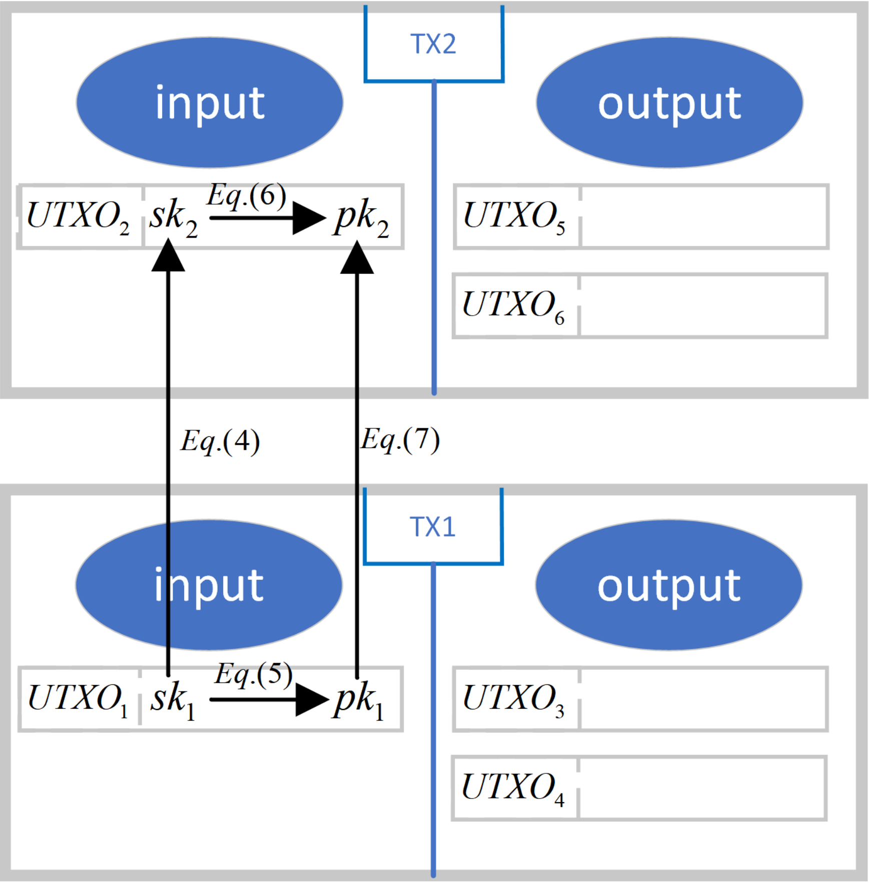


Figure 5. ECDH derivation.

Since the output addresses can be arbitrarily specified, there is no chain correlation between the input and unspent transaction outputs (UTXOs). Therefore, if a covert message is embedded in a set of transactions (TX_1, TX_2) generated according to Equations (4)–(7), these two transactions do not exhibit any explicit association. Except for the receiver who holds k_s, no one else can link these two transactions together, and thus, the complete nature of the covert transactions remains unknown.

5. The Proposed Scheme

In our proposed blockchain-based group covert communication scheme, a complete covert communication process consists of five steps, as shown in Figure 1. The sender needs to initialize and build covert transactions. The created covert transactions are propagated to the entire blockchain network through the flooding broadcast mechanism and recorded on the blockchain ledger. On the other hand, the receiver needs to filter covert transactions among all of the blockchain transactions and finally extract messages and verify them. The sequence of implementation and the algorithms are described in Figure 6.

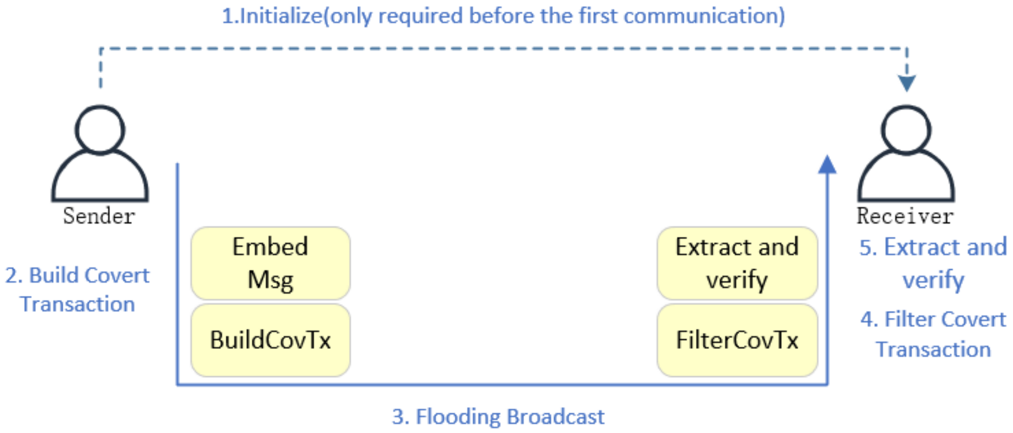

1.Initialize(only required before the first communication)
Sender
Receiver
2. Build Covert Transaction
Embed Msg
BuildCovTx
Extract and verify
FilterCovTx
5. Extract and verify
4. Filter Covert Transaction
3. Flooding Broadcast

Figure 6. The sequence of implementation.

5.1. Initialize

The sender generates pre-shared key k_s, message sequence number seq_n, a series of public–private key pairs including authentication key pairs (sk_m, pk_m) and signature embedding key pairs (sk_s, pk_s). These parameters are then transmitted to each receiver through the out-of-chain communication channel. Additionally, to facilitate the efficient filtering of covert transactions, the sender should inform receivers of the suitable starting block for the first covert transaction when establishing the covert channel. This step is only executed once during the initialization of communication and is not required afterwards.

5.2. Build Covert Transaction

The sender creates covert transactions that carry covert messages by executing the $EmbedMsg$ algorithm (Algorithm 3) and the $BuildCovTx$ algorithm (Algorithm 4). In this scheme, the sender needs to send a set of two covert transactions, denoted as $CTX = (tx_1, tx_2)$, in each time of covert communication. To simplify the notation, the parameters related to the transactions are represented with corresponding subscripts 1 or 2. The notations used in this section are illustrated in Table 1.

The sender splits the original covert message into two parts as $message = (message_1, message_2)$. Then, two random strings $random_1$ and $random_2$ are generated. They are concatenated with the current sequence number seq_n to form the message to be signed, which is denoted as $messageTosign$. The identity authentication signature σ_m is calculated by $Hash(messageTosign)$ and the identity authentication private key. In the proposed scheme, the identity authentication signature, as part of the embedded data, shares the embedding capacity with the actual message to be transmitted. RSA signatures are too long to be fully embedded in two covert transactions and are not suitable for the scheme. On the other hand, ECDSA signatures would encroach on the embedding space, greatly reducing the covert information transmission rate. Therefore, to maintain the confidentiality of transactions and an effective information transmission rate, we use a BLS short signature [5] with a total length of 33 bytes as the identity authentication signature, which has been proven to be sufficiently secure and reliable. The sender splits the identity authentication signature into two parts. σ_1 is concatenated with $random_1$ and $message_1$ to serve as random factor k_1 for signing the first transaction tx_1. σ_2 is concatenated with $message_2$, $random_2$, and seq_n to compose the sender's private key sk_2 for sending the second covert transaction tx_2. An example of the embedding format is shown in Figure 7. The lengths of each component can be adjusted according to the security and embedding rate requirements.

Table 1. List of notations.

Parameters	Description
sk	private key
pk	public key
k	random factor
$\|\|$	concatenating two strings
$sign_m()$	authentication signature algorithm
$sign_{ECDSA}()$	ECDSA signature algorithm
$Hash()$	hash function
$split()$	split a string
$txtosign$	unsigned transaction
σ	signature
$addr$	address
PRF	pseudo random function
(r,s)	signature value
$verify_{ECDSA}$	ECDSA signature verification algorithm
$verify_m$	authentication signature algorithm

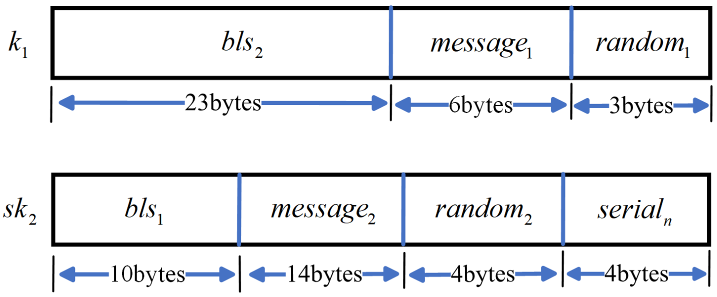


Figure 7. An example of embedding format.

Algorithm 3 EmbedMsg

```
Require: original text $message$, current sequence number $seq_n$, sender's authentication private key $sk_m$
Ensure: sender's private key $sk_2$ for sending $tx_2$, random factor $k_1$ for sending $tx_1$
Initialize $PRF$
Generate two random strings $random_1$ and $random_2$ using $PRF$
Set $(message_1, message_2) = split(message)$
Set $\sigma_m = sign_m(sk_m, Hash(messageTosign))$
Set $(\sigma_{m_1}, \sigma_{m_2}) = split(\sigma_m)$
Set $k_1 = \sigma_{m_1}||random_1||message_1$
Set $sk_2 = \sigma_{m_2}||random_2||seq_n$
return $sk_2, k_1$
```

After obtaining sk_2, the sender calculates the private key sk_1 for the first covert transaction based on the pre-shared random factor k_s. Next, the sender selects output addresses arbitrarily and constructs the unsigned transactions $tx_1tosign$ and $tx_2tosign$. The sender utilizes k_1 and private key sk_1 to sign $tx_1tosign$ for σ_1 by the ECDSA algorithm (refer to Section 4.2). Then, the signature σ_1, the address $addr_1$ corresponding to sk_1, and the unsigned transaction $tx_1tosign$ are combined together to generate the complete covert transaction tx_1.

When constructing the second covert transaction, the sender first initializes a pseudo-random function. As long as the same parameters are used for initialization, subsequent pseudo-random numbers generated in sequence will be identical. We use PRF to generate two pseudo-random numbers denoted as a and b in sequence. They are then involved in the calculation of the point (z_x, z_y) on the elliptic curve with k_1 and pk_s. The hash function is used on the x-coordinates z_x to obtain k_2. Next, the sender follows the steps of the ECDSA algorithm to generate σ_2 and build covert transaction tx_2. The sender can transmit either or both of the covert transactions to the blockchain at any time. The complete process of building covert transactions is illustrated in Figure 8.

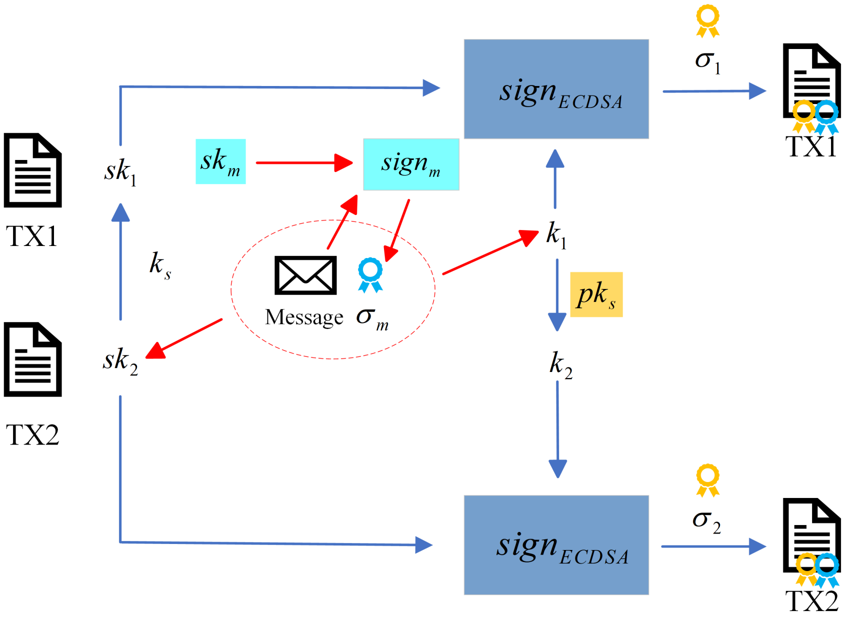


Figure 8. Build covert transactions.

Algorithm 4 BuildCovTx

```
Require: $sk_2, k_1$, pre-shared key $k_s$, embed public key $pk_s$, the generator point $G$
Ensure: a set of covert transaction $(tx_1, tx_2)$
Compute $sk_1 = sk_2 * k_s$
Generate $tx_1tosign$ and $tx_2tosign$
Set $\sigma_1 = sign_{ECDSA}(sk_1, k_1, tx_1tosign)$
Generate $addr_1$ corresponding to $sk_1$
Set $tx_1 = (addr_1, \sigma_1, tx_1tosign)$
Initialize $PRF$ with $tx_2tosign$
Generate two random factor $a$ and $b$ using $PRF$
Compute $(Z_x, Z_y) = G \cdot k_1 \cdot a + pk_s \cdot k_1 \cdot b$
Set $k_2 = Hash(z_x)$
Set $\sigma_2 = sign_{ECDSA}(sk_2, k_2, tx_2tosign)$
Generate $addr_2$ corresponding to $sk_2$
Set $tx_2 = (addr_2, \sigma_2, tx_2tosign)$
return $tx_1, tx_2$
```

5.3. Flooding Broadcast

Covert transactions are propagated to all Bitcoin nodes through the flooding broadcast mechanism in the same manner as normal transactions and eventually recorded on the blockchain. They are permanent stored and can be accessed at any given time. If both tx_1 and tx_2 can be confirmed in the blockchain, covert messages have been successfully sent.

5.4. Filter Covert Transaction

Receiver can filter out a set of covert transactions in a block through $FilterCovTx$ algorithm (Algorithm 5).

The receiver needs to identify a set of covert transactions $CTX = (tx_1, tx_2)$ sent by the sender from the blockchain. Initially, the receiver obtains a list of transactions TX_{list_2} in a block. The sender can inform the receiver of a suitable block number, allowing the receiver to quickly locate the block containing the covert messages in the first time of communication. For each transaction Tx_i in the list, the receiver extracts the sender's public key pk_i and then attempts to compute a public key pk_1 and its corresponding address $addr_1$ with k_s. The receiver then searches for all transactions associated with $addr_1$. If such transactions exist, they are potential candidates for the covert transactions sent by the sender and serve as the first covert transaction tx_1 of the covert transaction set CTX, while Tx_i serves as the matching tx_2. Generally, unless the sender deliberately sends other transactions from the address for the purpose of hiding, confusing or other reasons, there will usually be only one covert transaction that utilizes the address. If no transactions can be found, it indicates that the transaction is not a covert transaction, and the receiver continues to search for the next transaction in the list TX_{list_2} until all transactions have been filtered, which means that there are no covert transactions in the current block.

Algorithm 5 FilterCovTx

```
Require: a set $TX_{list_2} = (Tx_1, Tx_2, \ldots)$ that contains the receiver's newly obtained transaction, $k_s$
Ensure: the second covert transaction $tx_2$ and a covert transaction list $Tx_{list_1}$ matches with $tx_2$
for $Tx_i$ in $Tx_{list_2}$ do
    Extract $pk_i$ from $Tx_i$
    Compute $pk_1 = pk_i \cdot k_s$
    Generate $addr_1$ corresponding to $pk_1$
    Get $Tx_{list_1}$ according to $addr_1$
    if $Tx_{list_1}$ is not empty then
        Set $tx_2 = Tx_i$
        return $Tx_{list_1}, tx_2$
    end if
end for
return null
```

5.5. Extract and Verify

After the transactions are preliminary selected, the receiver extracts the covert messages among them using Algorithm 6 to verify the source and timeliness of the transactions.

First, the receiver extracts relevant transaction parameters, including the sender's signature, the sender's public key, and the unsigned transaction. Subsequently, for each transaction Tx_j in the list Tx_{list_1}, the receiver performs the ECDSA signature verification algorithm to obtain the point X. Next, the receiver initializes PRF with $tx_2tosign$ and generates two pseudo-random numbers denoted as a and b in sequence. The point (Z_x, Z_y) can be calculated by embedding private key sk_s. Then, the receiver performs a hash operation on the x-coordinate Z_x to obtain the random factor k_2 used for the second covert transaction's signature. With this, the private key d can also be derived. The receiver computes the corresponding public key, and if the result matches the actual public key pk_2, it further proves the successful pairing of a set of transactions $CTX = (tx_1, tx_2)$.

Algorithm 6 Extract and verify

```
Require: $Tx_{list_1}$, $tx_2$, authentication public key $pk_m$, the latest sequence number $seq_s$
Ensure: covert message $m$, updated sequence number $tx_n$
Parse $tx_2$ as $(r_2, s_2, pk_2, tx_2tosign)$
for $Tx_j$ in $Tx_{list_2}$ do
    Parse $Tx_j$ as $(r_j, s_j, pk_j, tx_jtosign)$
    Set $X = verify_{ECDSA}(r_j, s_j, pk_j, tx_jtosign)$
    Initialize $PRF$ with $tx_2tosign$
    Generate two random factor $a$ and $b$ using $PRF$
    Compute $(Z_x, Z_y = X \cdot a + X \cdot sk_s \cdot b)$
    Set $k_2 = Hash(Z_x)$
    Compute $d = r_2^{-1} * (k_2 * s_2 - Hash(tx_2tosign))$
    if $d \cdot G == pk_2$ then
        Compute $sk_1 = d * k_s$
        Compute $k_1 = s_j^{-1} * (Hash(tx_jtosign) + sk_1) * r_j$
        Parse $(d, sk_1)$ as $(\sigma_m, messageTosign, seq_n)$
        if $seq_n > seq_s$ and $verify_m(messageTosign, pk_m, \sigma m) ==$ True then
            Set $m = split(messageTosign)$
            return $m, seq_n$
        end if
    end if
end for
return null
```

To mitigate the possibility of transaction forgery and replay by internal adversary, the receiver needs to additionally verify the identity authentication signatures embedded in the transactions. The receiver first uses the recovered private key of the second covert transaction d and k_s to calculate the private key of the first covert transaction. Then, it recovers the random factor k_1 used during the transaction signing. The receiver can separate the identity authentication signature σ_m, *messageTosign*, and current sequence number seq_n according to the embedding rules. The transaction can be considered valid only if σ_m passes verification and satisfying $seq_n > seq_s$, which means the transaction is not being replayed. Finally, the receiver extracts the original covert message m embedded in the transaction, updates seq_s with seq_n and records the number of blocks containing tx_2. The next filtering process will begin from that block number.

6. Performance Analysis

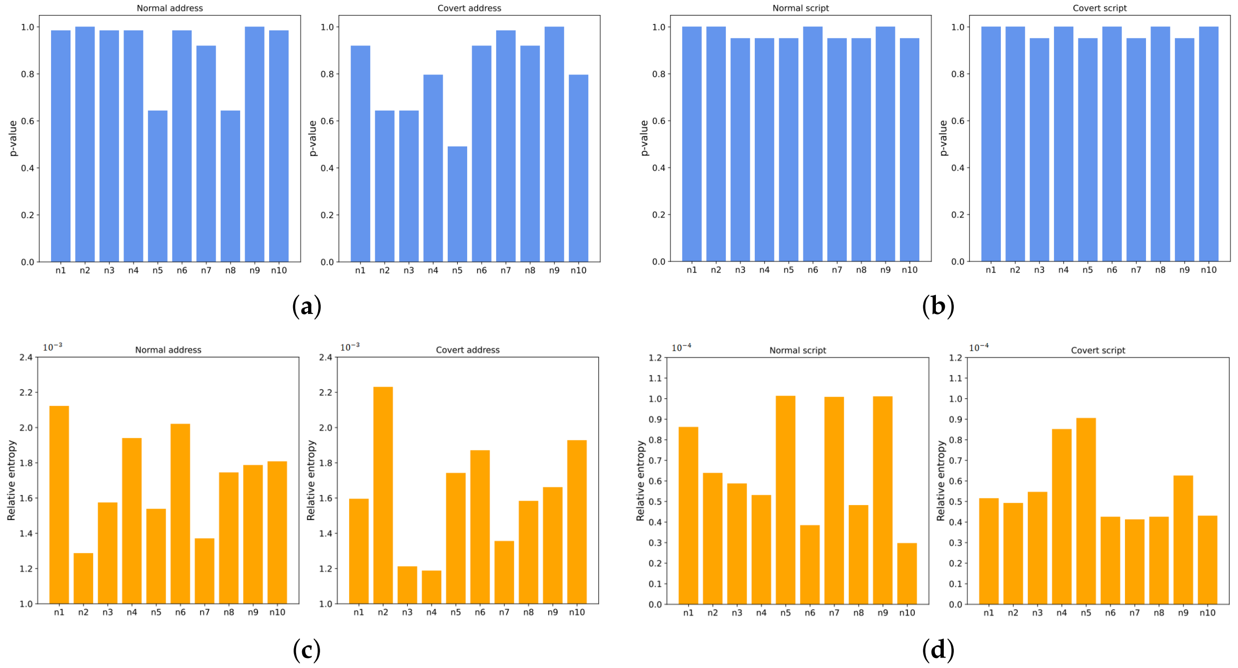
In this section, we present the experiment conducted on the prototype system and conduct an analysis of its performance toward various adversaries. The experimental results demonstrate that our scheme is more secure and undetectable compared to existing schemes with acceptable efficiency.

6.1. Concealment

Since the blockchain data analyst tends to discover covert channels by statistical distribution, we employ the Kolmogorov–Smirnov (KS) test and the Kullback–Leibler divergence (KLD) test [22], which are two types of detection metrics to demonstrate the resilience of the proposed scheme against statistical analysis. The KS test utilizes the p-value to determine the difference between two sample distributions. When $p > 0.05$, it can be considered that the two samples follow the same distribution and there is no significant difference between them. KLD, also known as relative entropy, quantifies the difference between two sample distributions. When the two distributions are totally identical, the KLD result is 0. Otherwise, a larger KLD value indicates a greater difference. Therefore, if the difference between hidden transactions and normal transactions is smaller than or approximately equal to the difference threshold of normal transactions, it indicates that covert transactions cannot be distinguished from normal transactions.

We download 10,000 normal transactions from the Bitcoin main network. These transactions were then randomly divided into 10 groups, each containing 1000 transactions, to form a normal transaction dataset denoted as $NG = (n1, n2, \ldots, n9, n10)$. We also download an additional 1000 normal transactions to build a baseline group, which represents the character distribution features of normal transactions. Furthermore, using the proposed approach, we generated 10,000 special transactions. Similar to the normal transaction dataset, these special transactions were randomly divided into 10 groups, each containing 1000 transactions to build a covert transaction dataset denoted as $CG = (c1, c2, \ldots, c9, c10)$. Since the differences are reflected only in the sender's address and script fields, we collect transaction data in these two fields for the KS test and KLD test. In conclusion, we have obtained the test results for each group compared to the baseline group.

As shown in Figure 9a,b, in the KS test for the sender's address and script of normal transactions, all the KS p-values are greater than 0.05. This suggests that the address and script fields of normal transactions conform to a specific distribution, meeting the conditions for using the KS test. Meanwhile, the KS p-values for the covert transactions are also found to be greater than 0.05, implying that they follow the same distribution as the corresponding fields in normal transactions and are indistinguishable.



(**a**) (**b**)

(**c**) (**d**)

Figure 9. KS and KLD tests. (**a**) KS test for address, (**b**) KS test for script, (**c**) KLD test for address, (**d**) KLD test for script.

Figure 9c displays the results of the KLD test for the address field. It can be observed that all of the values in the normal transaction groups are below 2.4×10^{-3}. As long as all KLD values of covert transactions are below this threshold, the statistical distributions of covert transactions and normal transactions can be ignored. Hence, we set the KLD threshold for the address field as $\theta_{addr} = 2.4 \times 10^{-3}$. All of the KLD test values for the covert transaction group are below θ_{addr}. This signifies that the differences between all special transaction samples and normal transaction samples cannot be distinguished from the differences among normal transaction samples alone. Thus, the address field of covert transaction can pass KLD tests and possesses significant concealment capabilities.

Similarly, in Figure 9d, the KLD threshold for the script field is set to $\theta_{script} = 1.1 \times 10^{-4}$ for normal transactions. All of the KLD values of covert transactions are less than θ_{script}, which proves that the script field of covert transaction is able to resist the KLD test and is indistinguishable from other transactions.

In addition to statistical analysis, we refer to Wang's Covert Transaction Recognition model [23], which is proposed by the Text Convolutional Neural Networks and Back Propagation Neural Networks, to identify the proposed covert transactions. The proximity of precision, recall and F_SCORE to 0.5 suggests the challenge of differentiating regular transactions from covert ones using machine learning detection methods. This trend underscores the enhanced concealment of covert transactions. In scenarios employing the OP_RETURN field [6,24], these schemes exhibit precision, recall and F_SCORE values all exceeding 0.99, signifying their high non-covert nature. Alternatively, schemes that directly encode information into other adjustable fields, such as amount [4] and output address [25,26], maintain precision, recall and F_SCORE levels above 0.7. Comparatively, the DSA [14] scheme registers precision, recall and F_SCORE at 0.54, 0.53, and 0.54, respectively, making it the most covert among these schemes due to its values closely approximating 0.5. The result of our scheme is shown in Table 2. The identified precision and recall of input addresses is 0.57, while the input script is 0.59. When considering both factors simultaneously, the result is 0.62. This indicates that increasing the embedding locations results in a slight decline in concealment. However, overall, our proposed scheme exhibits performance that is closest to the DSA scheme, outperforming all other schemes. In other words, our scheme demonstrates a distinct advantage in terms of concealment.

Table 2. The result of covert transaction recognition (embedded field).

	Precision	Recall	F_Score
Our scheme's input addresses	0.57	0.57	0.57
Our scheme's input script	0.59	0.59	0.59
Our scheme's input addresses and script	0.62	0.62	0.61

Referring to Section 5, correlations between transactions can compromise concealment. In our proposed scheme, all addresses that have been used are utilized only once, thus eliminating static-address correlation. There is no sequential linkage in historical transactions, thus preventing a chain-address correlation between covert transactions. The temporal occurrence of all covert transactions remains unpredictable, even for internal adversaries, who are unable to anticipate the addresses employed in transactions. The difference between the covert transactions created by the proposed scheme and regular transactions lies solely in the parameters used during transaction generation. These parameters are not influenced by time or space. Therefore, adversaries cannot distinguish between them based on factors such as time intervals, frequencies, or other patterns. Our proposed scheme exhibits sufficient concealment to withstand detection by various adversaries.

Even if an adversary understands the algorithm, they can only attempt to find covert transactions by exhaustively trying different k_s values. Specifically, the adversary exhaustively searches through k_s values, starting from any address, and calculates the addresses that are linked to that address. If such addresses exist, they could potentially be used for covert transactions. The range of k_s values is 256 bits, and the sender randomly selects k_{s_i} values to represent group i from this range. As the number of groups increases, the effective range of k_s values gradually expands. Now, consider an adversary randomly generating their own k_c, with the current number of groups being g_s. The collision probability of k_c being an effective k_s value is $\frac{g_s}{2^{256}}$. For example, when the number of groups is 1000, the collision probability is approximately 8.6×10^{-75}.

6.2. Practicability

We implement our scheme on an Alibaba cloud server running a Ubuntu 20.04 x64 operating system with Intel (R) Xeon (R) CPU E5-2682 v4 @ 2.50GHz and 8 GB RAM. The average time consumption of each process is shown in Table 3.

Table 3. Average time consumption of each process.

Process	Normal Transaction (s)	Covert Transaction (s)
Embed messages	3.2679×10^{-6}	0.0133
Build transaction	0.0314	0.0341
Filter transaction	2.3409×10^{-5}	0.0021
Extract and verify	null	0.0251

In contrast to normal transactions, embedding messages in covert transaction takes more time consumption because k and sk are customized rather than random. The identity authentication signature also needs time to be generated. When building transactions, since the rules of arithmetic have not changed, there is little difference in time cost. Combining these two steps, sending a covert transaction takes about 0.04752 s, while sending a normal transaction takes about 0.03141 s with an instance of only 0.01611 s, which is not significantly different.

We download 5000 transactions from the Bitcoin main network to test the time required to filter transactions. The ratio of normal transactions to covert transactions is 5000:1. As shown in Table 3, on average, the time consumption of the filtering transaction is around 0.00212 s per transaction. Since transactions are sequentially scanned, the time cost is linearly related to the size of the transactions set n. The time complexity is $O(n)$. When n is 5000, the total time taken is 10.5947 s. Bitcoin typically generates a new block in approximately 10 min, and the number of transactions in a block is usually less than 5000. Therefore, in around 10 s, the receiver can complete the traversal of a block. The filtering speed is 60 times faster than the speed of new block generation. This means that the receiver does not need to synchronize and maintain block numbers incessantly. Filtering can be started from any block at any moment and is able to identify transactions in the latest blocks.

6.3. Unforgeability

The transactions embed identity authentication signatures, and only the genuine sender possesses the corresponding private key, rendering internal adversaries incapable of generating valid signatures. Therefore, even with knowledge of the covert transaction generation algorithm, the forging of entirely new transactions is unattainable. The signature encompasses the message and sequence number. Any modification to either element results in the failure of the original signature verification. Consequently, the receivers can discern that a forged transaction originates from an internal adversary rather than the authentic sender. Thus, adversaries are precluded from fabricating transactions by replaying past transactions or duplicating previous signatures. In summary, the transactions generated by our group covert communication scheme exhibit unforgeability.

6.4. Capability Comparison

To facilitate the comparison of concealment, channel capacity, transmission efficiency, and transaction fees among different schemes, we adopted a single-input and two-output mode for each transaction. Firstly, as of April 2023, this transaction mode accounted for almost 50% [19], making it more universally applicable and providing inherent concealment for covert transactions. Secondly, a single transaction can have multiple inputs and outputs, which means that multiple transaction fields, such as output addresses and signatures, can be constructed to embed covert information. Theoretically, the amount of information embedded in a single transaction can be limitless. Therefore, it is necessary to specify the number of inputs and outputs to compare the embedding rate while ensuring the same level of concealment.

The characteristics of the P2PKH-type transactions generated by each covert communication scheme are presented in Tables 4 and 5. The static-address correlation and chain-address correlation have been discussed in Section 4.1.2, which make the channel susceptible to address clustering analysis attacks [27]. We set the transaction fee as 0.0000747 BTC/KB acccording to block 801,087 [28], and the exchange rate between Bitcoin and the US Dollar is set as 1:29,325.

Table 4. Comprehensive performance (concealment).

Scheme	Minimum TX for Communication	Maximum Times for Address Reuse	Static-Address Correlation	Chain-Address Correlation	Explicit Embedding	Resist Internal Adversary
BLOCCE [25]	1	always	√	√	×	√
Zombiecoin [10]	1	always	√	√	√	√
Chainchannel [13]	2	2	√	×	×	√
DLchain [15]	2	2	√	×	√	×
ECDHC-CDE [16]	1	1	×	√	×	√
Kleptography [6]	2	always	√	√	√	×
Digital Currency [4]	1	1	×	×	√	×
Our scheme	2	1	×	×	×	√

Compared to other schemes, our scheme exhibits the highest level of concealment as it possesses neither reuse addresses nor any address correlation. Additionally, it does not utilize the OP_RETURN field for explicit embedding. Consequently, it ensures maximum concealment. Moreover, it maintains an embedding rate of 10 bytes per transaction with a cost of 0.122 USD/bit while countering internal adversaries with acceptable filtering efficiency of $O(n)$.

Table 5. Comprehensive performance (capacity and cost).

Scheme	Channel Capacity per TX	Channel Capacity per Covert Communication	TX Size per Covert Communication	Embedding Rate	Cost/bit (Satoshi)	Cost/Bit (USD)	Filter Efficiency
BLOCCE [25]	1 bit	1 bit	225 bytes	0.056%	1681	0.4929	$O(1)$
Zombiecoin [10]	80 bytes	80 bytes	315 bytes	25.397%	4	0.0011	$O(1)$
Chainchannel [13]	15 bytes	30 bytes	450 bytes	6.667%	14	0.0041	$O(1)$
DLchain [15]	16 bytes	32 bytes	630 bytes	5.079%	18	0.0054	$O(n^2)$
ECDHC-CDE [16]	1 bit	1 bit	225 bytes	0.056%	1681	0.4929	$O(1)$
Kleptography [6]	80 bytes	160 bytes	630 bytes	25.397%	4	0.0011	$O(1)$
Digital Currency [4]	14 bits	14 bits	225 bytes	6.222%	120	0.0352	$O(1)$
Our scheme	10 bytes	20 bytes	450 bytes	4.444%	21	0.0062	$O(n)$

6.5. Prevention of Malicious Adversary

- Prevention of Network Eavesdropper: Network eavesdroppers attempt to capture and analyze the traffic generated during the data transmission process. The off-chain channel is only used once during the initial establishment of communication, significantly reducing the possibility of data leakage. On the other hand, data transmission within the blockchain utilizes a flooding broadcast mechanism, where the packets exchanged between nodes only contain transaction data itself. The source and destination addresses within these packets represent node addresses rather than the addresses of the sender and receiver in covert communication. Furthermore, the data format of covert transactions is identical to that of normal transactions. As a result, network eavesdroppers are unable to identify the identities of the communicating parties through the analysis of network traffic data.
- Prevention of blockchain data analyst: The fields influenced by covert messages can resist statistical analysis methods like the KS test and KLD test. The character distribution of these fields cannot be distinguished from normal transactions. The generation method of the other fields is exactly the same as that of normal transactions. Therefore, a blockchain data analyst cannot differentiate between covert transactions and normal transactions. On the other hand, covert transactions are independent from each other, making their construction patterns identical to normal transactions. As a

result, adversaries are unable to confirm the existence of covert channels based on the correlation between covert transactions.

- Prevention of internal adversary: In our scheme, the private key to create covert transactions is exposed on the receiver's side. Therefore, the members of the receiver group also have the capability to construct covert transactions. However, due to the identity authentication signatures embedded in the transactions, as long as the corresponding private keys for the signatures are kept absolutely confidential, malicious receivers are unable to generate correct identity authentication signatures for covert transactions they forge. As a result, their attempt to exploit this channel and steal the sender's identity for communication with other receivers cannot be achieved.

  On the other hand, the identity authentication signature includes a sequence number of the message which strictly increases, implying a unique correspondence between each sequence number and the historical messages. An internal adversary is unable to pass the sequence number authentication by completely replaying the covert transactions previously constructed by the sender. Consequently, the forged transactions will not be accepted by other receivers. Moreover, this abnormal behavior would expose the presence of an internal adversary within the receiver group. Based on these premises, we can conclude that rational malicious receivers would refrain from forgery and replay attacks. The internal adversary is well acquainted with the characteristics and patterns of covert transactions, which may increase their accuracy in identifying covert transactions transforming to other groups. Machine learning detection, based on known features of covert transactions, can be used to search for other covert transactions and simulate the behavior of internal adversary. The experimental results demonstrate that even with the adversary's familiarity with the scheme and covert transaction features, our proposed scheme still maintains concealment.

7. Conclusions and Future Work

The decentralized and flood broadcasting nature of a blockchain enables communication parties to achieve group covert communication without the usage or association of real identity information. However, existing schemes based on blockchain transactions exhibit significant transaction correlations, posing the risk of being traced. Additionally, these schemes lack countermeasures against malicious receivers while ensuring both covert operation and sufficient transmission efficiency, resulting in the reduced reliability of the channel.

This paper presents a blockchain-based group covert communication scheme. We improve ECDH and the concept of kleptography to achieve a higher concealment of covert transactions. The covert transactions generated by this scheme exhibit no correlations and thus offer untraceability and unlinkability. We also attempt to embed a short identical authentication signature to prevent attacks from internal adversaries. Consequently, our scheme can resist statistical analysis attacks and machine learning analyses. The precision and recall of machine learning detection result can reach 0.57–0.62 (0.5 is the ideal value); thus, they are proved to be undetectable. In comparison to other group covert communication schemes, our scheme not only considers filtering efficiency and channel capacity but also resolves replay attacks and identity forgery attacks from internal adversaries. This further enhances the reliability of communication.

The objective of this paper is to prevent internal adversaries from successfully executing replay and forgery attacks while acknowledging the limitation of countering constant monitoring. To further mitigate the threats posed by internal adversaries, it is necessary to explore effective methods for dynamically updating group members and removing specified malicious receivers from the group.

Author Contributions: Conceptualization, T.S.; Methodology, T.S.; Software, L.Z. and F.G.; Validation, Z.C. and F.G.; Formal analysis, Z.Z. and M.L.; Investigation, F.G. and L.Z.; Writing—original draft, T.S., M.L. and Z.Z.; Writing—review and editing, T.S. and M.L. and Z.Z.; Visualization, L.Z.; Data curation, Z.C. and F.G.; Supervision, Z.Z. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Defense Basic Scientific Research program of China under grant number JCKY2020602B008.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, T.; Li, B.; Zhu, Y.; Han, T.; Wu, Q. Covert channels in blockchain and blockchain based covert communication: Overview, state-of-the-art, and future directions. *Comput. Commun.* **2023**, *205*, 136–146. [CrossRef]
2. Caviglione, L. Trends and challenges in network covert channels countermeasures. *Appl. Sci.* **2021**, *11*, 1641. [CrossRef]
3. Giron, A.A.; Martina, J.E.; Custódio, R. Steganographic analysis of blockchains. *Sensors* **2021**, *21*, 4078. [CrossRef] [PubMed]
4. Zhang, P.; Cheng, Q.; Zhang, M.; Luo, X. A group covert communication method of digital currency based on blockchain technology. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 4266–4276. [CrossRef]
5. Boneh, D.; Lynn, B.; Shacham, H. Short signatures from the weil pairing. In *Advances in Cryptology—ASIACRYPT 2001, Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, 9–13 December 2001*; Proceedings 7; Springer: Berlin/Heidelberg, Germany, 2001; pp. 514–532.
6. Gao, F.; Zhu, L.; Gai, K.; Zhang, C.; Liu, S. Achieving a covert channel over an open blockchain network. *IEEE Netw.* **2020**, *34*, 6–13. [CrossRef]
7. Lampson, B.W. A note on the confinement problem. *Commun. ACM* **1973**, *16*, 613–615. [CrossRef]
8. Trabelsi, Z.; El-Hajj, W.; Hamdy, S. Implementation of an icmp-based covert channel for file and message transfer. In Proceedings of the 2008 15th IEEE International Conference on Electronics, Circuits and Systems, Saint Julian's, Malta, 31 August–September 2008; pp. 894–897.
9. Gianvecchio, S.; Wang, H. An entropy-based approach to detecting covert timing channels. *IEEE Trans. Dependable Secur. Comput.* **2010**, *8*, 785–797. [CrossRef]
10. Ali, S.T.; McCorry, P.; Lee, P.H.-J.; Hao, F. Zombiecoin: Powering next-generation botnets with bitcoin. In *Financial Cryptography and Data Security: FC 2015, Proceedings of the International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, 26–30 January 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 34–48.
11. Bartoletti, M.; Pompianu, L. An analysis of bitcoin op_return metadata. In *Financial Cryptography and Data Security: FC 2017, Proceedings of the International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, 3–7 April 2017*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 218–230.
12. Liu, F.; Li, Z.; Jia, K.; Xiang, P.; Zhou, A.; Qi, J.; Li, Z. Bitcoin address clustering based on change address improvement. *IEEE Trans. Comput. Soc. Syst.* **2023**, 1–12. [CrossRef]
13. Frkat, D.; Annessi, R.; Zseby, T. Chainchannels: Private botnet communication over public blockchains. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1244–1252.
14. Fionov, A. Exploring covert channels in bitcoin transactions. In Proceedings of the 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 21–27 October 2019; pp. 59–64.
15. Tian, J.; Gou, G.; Liu, C.; Chen, Y.; Xiong, G.; Li, Z. Dlchain: A covert channel over blockchain based on dynamic labels. In Proceedings of the Information and Communications Security: 21st International Conference, ICICS 2019, Beijing, China, 15–17 December 2019; Revised Selected Papers 21; Springer: Berlin/Heidelberg, Germany, 2020; pp. 814–830.
16. Cao, H.; Yin, H.; Gao, F.; Zhang, Z.; Khoussainov, B.; Xu, S.; Zhu, L. Chain-based covert data embedding schemes in blockchain. *IEEE Internet Things J.* **2020**, *9*, 14699–14707. [CrossRef]
17. Zhang, Z.; Zhang, L.; Rasheed, W.; Jin, Z.; Ma, T.; Chen, H.; Xu, G. The research on covert communication model based on blockchain: A case study of ethereum's whisper protocol. In *Frontiers in Cyber Security, Proceedings of the Third International Conference, FCS 2020, Tianjin, China, 15–17 November 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 215–230.
18. Baden, M.; Torres, C.F.; Pontiveros, B.B.F.; State, R. Whispering botnet command and control instructions. In Proceedings of the 2019 Crypto Valley Conference on Blockchain Technology (CVCBT), Rotkreuz, Switzerland, 24–26 June 2019; pp. 77–81.
19. Transaction Fee Information. Available online: https://transactionfee.info/ (accessed on 5 December 2023).
20. Young, A.; Yung, M. Kleptography: Using cryptography against cryptography. In *Advances in Cryptology—EUROCRYPT'97, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, 11–15 May 1997*; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 1997; pp. 62–74 .

21. Genç, Y.; Afacan, E. Design and implementation of an efficient elliptic curve digital signature algorithm (ecdsa). In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021; pp. 1–6.
22. Xiang, L.; Wang, R.; Yang, Z.; Liu, Y. Generative linguistic steganography: A comprehensive review. *Ksii Trans. Internet Inf. Syst.* **2022**, *16*, 986–1005.
23. Wang, M.; Zhang, Z.; He, J.; Gao, F.; Li, M.; Xu, S.; Zhu, L. Practical blockchain-based steganographic communication via adversarial ai: A case study in bitcoin. *Comput. J.* **2022**, *65*, 2926–2938. [CrossRef]
24. Matzutt, R.; Hiller, J.; Henze, M.; Ziegeldorf, J.H.; Müllmann, D.; Hohlfeld, O.; Wehrle, K. A quantitative analysis of the impact of arbitrary blockchain content on bitcoin. In Proceedings of the Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, 26 February–2 March 2018; Revised Selected Papers 22; Springer: Berlin/Heidelberg, Germany, 2018; pp. 420–438.
25. Partala, J. Provably secure covert communication on blockchain. *Cryptography* **2018**, *2*, 18. [CrossRef]
26. Ali, S.T.; McCorry, P.; Lee, P.H.-J.; Hao, F. Zombiecoin 2.0: Managing next-generation botnets using bitcoin. *Int. J. Inf. Secur.* **2018**, *17*, 411–422. [CrossRef]
27. Saxena, R.; Arora, D.; Nagar, V. Efficient blockchain addresses classification through cascading ensemble learning approach. *Int. J. Electron. Secur. Digit. Forensics* **2023**, *15*, 195–210. [CrossRef]
28. Block Information. Available online: https://www.blockchain.com/explorer/blocks/btc/801087 (accessed on 5 December 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.