Security and Safety, Vol. 3, 2024006 (2024)
https://doi.org/10.1051/sands/2024006
sands.edpsciences.org

Security and Privacy for Space-Air-Ground Integrated Networks

Research Article

OPEN ACCESS

Information Network

Secure and efficient covert communication for blockchain-integrated SAGINs

Weijia Li, Yuan Zhang*, Xinyu He, and Yaqing Song

School of Computer Science and Engineering (School of Cyber Security), University of Electronic Science and Technology of China, Chengdu 611731, China

Received: 30 March 2024 / Revised: 24 April 2024 / Accepted: 28 April 2024 / Published online: 30 April 2024

Abstract Blockchain has brought great potential in improving Space-Air-Ground Integrated Networks (SAGINs) in terms of security and efficiency. In blockchain-integrated SAGINs, many applications and services inherently require both the communication contents and communication behaviors to be secure against eavesdroppers, in which a covert communication algorithm is always deployed as a fundamental communication component. However, existing covert communication schemes suffer from critical problems. On the one hand, they require a sender to locally maintain a cryptographic key for a long period of time, which is very costly and inefficient to renew which means renewing the secret key. On the other hand, the ciphertext of covertly sent data would explicitly appear in the network, and thereby the schemes are vulnerable to secret key breach. In this paper, we propose a secure and efficient covert communication scheme for blockchain-integrated SAGINs, dubbed CC-BSAGINs, to free the sender from maintaining secret keys. The key technique is to map the covertly sent data to some transactions on the underlying blockchain in a secure and efficient way; the mapping information is sent *via* a covert communication algorithm. Such a two-step mechanism releases the sender from key management and does not require the ciphertext to be communicated. We provide formal security proofs and conduct a comprehensive performance evaluation, which demonstrates the security and efficiency of CC-BSAGINs.

Keywords Covert communication, blockchain, Space-Air-Ground Integrated Networks

Citation Li W, Zhang Y, He X, and Song Y. Secure and efficient covert communication for blockchain-integrated SAGINs. Security and Safety 2024; **3**: 2024006. https://doi.org/10.1051/sands/2024006

1 Introduction

Space-Air-Ground Integrated Networks (SAGINs) have gained significant attention and become a promising architecture for ubiquitous connectivity 5G-Advanced and 6G, enabling the integration of satellite networks, aerial networks, and terrestrial networks. This integration brings tremendous communication benefits, such as non-terrestrial networks, seamless global coverage, high flexibility, and augmented system capacity [1]. SAGINs can be regarded as an extension of the traditional network, which has a strong demand for secure and reliable communication, especially in extreme environments [2].

Generally, the reliability and security of SAGINs are guaranteed by utilizing cryptographic primitives, *e.g.*, public/symmetric-key encryption and digital signatures. However, critical issues in terms of security and efficiency still exist in deploying the primitives in SAGINs. Regarding security, most of the existing public-key encryption schemes and signatures rely on public key infrastructure (PKI), where a

* Corresponding author (email: ZY_LoYe@126.com)

fully trusted certificate authority (CA) is required to issue a certificate for each entity. As a consequence, CA becomes a single point of failure, and adversaries who compromise CA can break the security of the underlying primitive. Regarding efficiency, PKI-based schemes are confronted with certificate management problems, including certificate revocation, storage, distribution, and verification. It would be very costly for application scenarios where the entities are dynamic and updated frequently. The above issues would be further exacerbated in deploying the PKI-based schemes in SAGINs, due to the complexity of SAGINs.

Blockchain can serve as a key complement to address the above problems. Specifically, as a blockchain system provides a publicly verifiable and tamper-resistant database, the certificate of each user can be recorded in it to ensure authenticity and the single-point-of-failure problem can be addressed [3, 4]. Such a technique has been deployed in SAGINs [5–7] and brought great potential in improving SAGINs in terms of security and efficiency.

In addition to the improvement of security and efficiency, integrating blockchain into SAGINs also provides a "new" way to achieve the "traditional" goal. Particularly, in some applications of SAGINs, users' communication behaviors are as sensitive as their communication content and thereby need to be well protected. Traditionally, users always utilize a *covert communication* scheme to protect their communication behaviors against adversaries. However, it always requires an underlying application service to "parasitize", which always causes abnormal communications. In blockchain-integrated SAGINs, covert communication can be achieved by accessing blockchain-related services for users: anyone who can access the blockchain can send/receive the message in a secure and covert way. Typical works include Ref. [8–10]. Despite the great benefits of blockchain-based covert communication schemes, there are also critical issues in terms of security and efficiency. Specifically, in existing schemes [11–13] senders need to well maintain *cryptographic secret keys* for a long period, and a message containing the covertly sent data is sent to the receiver. Consequently, if a sender is captured by adversaries, not only the communication behavior but also the communication content would be directly leaked. A straightforward way to mitigate this problem is to frequently update the secret key. However, it would introduce prohibited costs on the sender side, as generating, updating, and distributing cryptographic keys are very cumbersome, especially for SAGINs where the users' devices are always resource-constrained. Although some works have been proposed to improve the efficiency of blockchain-based covert communication, the fundamental issue of maintaining cryptographic secret keys on the sender is still not resolved.

In this paper, we propose an efficient covert communication scheme for blockchain-integrated SAGINs, dubbed CC-BSAGINs, which frees the sender from maintaining secret keys. Specifically, CC-BSAGINs utilizes a two-step paradigm: in the first step, a sender transfers the covertly sent data to a "treasure map"; in the second step, the sender sends the treasure map (rather than the ciphertext of covertly sent data) to the receiver in a covert and secure way using another covert communication algorithm. The treasure map is instantiated by utilizing the underlying blockchain of SAGINs in tandem with an efficient index mechanism. By doing so, the ciphertext of covertly sent data would not appear in the network, a sender just needs to maintain a "transformation" algorithm, which can be updated after each communication for security reasons and does not require the sender to maintain any secret key locally. Furthermore, CC-BSAGINs are compatible with existing covert communication schemes and would inherit all the features. Specifically, the contributions of this work are summarized as follows.

(1) We propose a two-step paradigm of covert communication, where the ciphertext of covertly sent data would not appear in the network, and the receiver can extract the data from a secure transformation mechanism. We also instantiate the transformation using blockchain and an efficient index algorithm, where only lightweight cryptographic operations, *e.g.*, hash function and comparison, are involved.
(2) We integrate the above mechanism into a covert communication scheme and develop the system, dubbed CC-BSAGINs, in blockchain-integrated SAGINs, which frees the sender from maintaining long-term cryptographic secret keys and ensures the data confidentiality even if the sender is controlled by the adversary.
(3) We provide formal security proofs and conduct a comprehensive performance evaluation, which demonstrates that CC-BSAGINs are secure and efficient.

The remainder of this paper is organized as follows. We review the related works in Section 2 and introduce the preliminaries in Section 3. We propose CC-BSAGINs in Section 4 and analyze the security in Section 5. In Section 6, we conduct a performance evaluation. Finally, we conclude and look at the future work in Section 7.

2 Related works

Covert communication can be traced back to the steganography technique of the 16th century, and the core idea is to hide communication messages using a physical or chemical method known only to a receiver. Covert communication is commonly depicted using the Prisoner's Dilemma proposed by Simmons [14]. It can be succinctly described as follows: Alice and Bob are inmates who seek to escape from prison, yet all their communications are under strict surveillance by the prison warden, Willie. Any suspicious behavior detected by him would result in harsher penalties for them. In the second half of the 20th century, with the advent of the communications Internet, covert communication schemes are often constructed using communications and Internet technology. The core is to embed the covert data in the redundant information of the ordinary transmitted data. For example, error-correcting codes are often used as carriers to store covert information in short-wave and satellite communications. In addition, images and videos are often used for covert transmission of data [15–22]. Ma *et al.* proposed a novel method by reserving room before encryption with a traditional reversible data hiding (RDH) algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image [23]. Sharifzade *et al.* [24] proposed a novel Gaussian embedding model by maximizing the detection error of the most common optical detectors within the adopted statistical model. They also extended the formulation to a cost-based steganography, resulting in a universal embedding scheme that improves the empirical results of current cost-based and statistical model-based approaches.

With the development of modern cryptography technology, a large number of covert communication schemes using cryptography have emerged. The core of these schemes is to embed covert data into digital signatures. Simmons constructed the first scheme using cryptographic techniques for covert communication, which successfully constructed a covert channel in the DSA [25]. Moreover, it is proved that there is also a covert channel in the ElGamal signature [26] and the ECDSA [27]. Anderson *et al.* found a class of covert channels in the ElGamal signature that combined the advantages of wideband and narrowband channels, that is, both the security of narrowband and wide bandwidth [26]. Jan *et al.* proposed two covert communication schemes based on discrete logarithms that shorten the length of required keys and digital signatures [28]. These two schemes may contain two or more covert messages in the signature, corresponding to different covert receivers, which shorten the required key and the length of the digital signature. Hartl *et al.* showed the existence of a broadband covert channel in the EdDSA [29] signature scheme [30]. Then they discussed the implications of the covert channel in practice using three different scenarios: broadcast clock synchronization, signed sensor data export, and classic TLS.

However, these schemes still have the problem of weak concealment of communication behaviors: the transmission of covert data depends on the generation of digital signatures, which may make adversaries notice the existence of covert channels.

To solve this problem, we started to build covert communication schemes using blockchain [31–37], because each transaction on the blockchain needs to generate a digital signature. The blockchain has the properties of anti-destruction and persistent storage, and cannot be tampered with. Alsalami *et al.* drew attention to the potential threat of abusing uncontrolled randomness in blockchain cryptographic algorithms [38]. They proposed a new steganographic technique that affects most cryptocurrencies. Based on the novel blockchain steganographic technique, they designed and implemented a practical covert communication system. Cao *et al.* [39] proposed a hash chain-based covert data embedding (HC-CDE) scheme. Besides, they proposed an elliptic curve Diffie-Hellman chain-based covert data embedding (ECDHC-CDE) scheme to enhance the security of the HC-CDE scheme. Luo *et al.* [10] proposed a covert communication method based on Bitcoin transactions.

Chen *et al.* did an extensive survey to investigate many covert communication schemes built on top of blockchain [40]. Gao *et al.* proposed a covert communication scheme for blockchain [41], which uses kleptography technology [42] to achieve high concealment and high-performance data transmission in an open network environment. Tian *et al.* proposed a blockchain covert channel construction scheme DLChain [43], in which dynamic labels were used instead of fixed labels to identify transactions containing covert information, and a dynamic label generation algorithm based on the statistical distribution of actual transaction data was designed to ensure the invisibility of dynamic labels. Zhang *et al.* proposed a covert communication method based on secret sharing and STC mapping on the public chain [44]. The method used the mapping relationship and transaction amount intertwined to complete the transmission

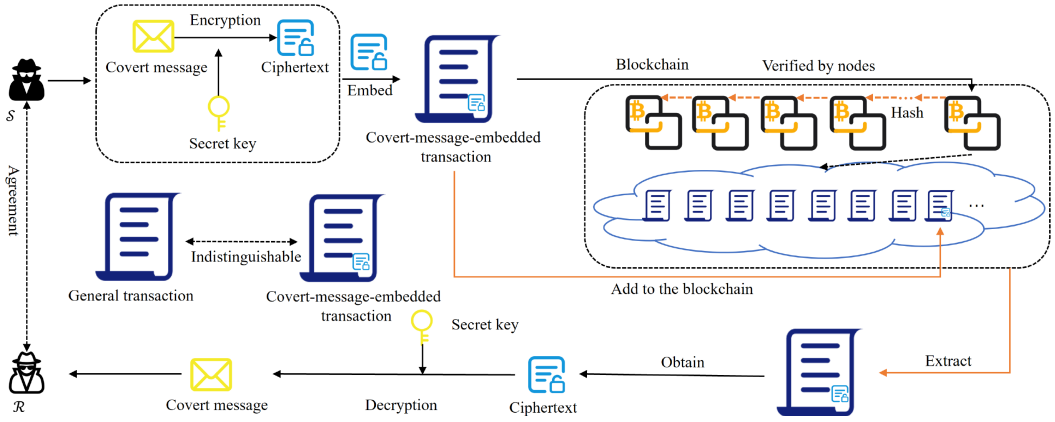

Figure 1. Blockchain-based covert communication model

Table 1. Comparison of existing blockchain-integrated covert communication schemes with CC-BSAGINs

Scheme	[41]	[44]	[30]	[46]	[48]	CC-BSAGINs
Confidentiality of messages	Yes	Yes	Yes	Yes	Yes	Yes
Concealment of communication behaviors	No	Yes	No	No	/	Yes
Anonymity of communication entities	Yes	Yes	No	No	Yes	Yes
Computational costs	24.8 ms	/	1s	/	3 s	0.3828 s
Communication costs	2 MB	$\leq$50 B	310 B	3 KB	/	50.5 KB
Cryptocurrency costs	$ 0.002	$ 0.3	0	/	$ 0.01	0
Key management issues	No	Yes	Yes	Yes	/	No
Ciphertext leakage issues	Yes	Yes	Yes	Yes	Yes	No

of secret information and thus achieved covert communication. Basuki *et al.* proposed a smart contract-based covert channel coding SCCCE scheme [45] and combined it with the image steganography algorithm to realize a covert sending of private data for Ethereum. In this scheme, the data to be transmitted is embedded into the image, and then the URL of the image is embedded into the transaction. By using image steganography, the amount of data that can be embedded is greatly increased. Liu *et al.* used the VALUE field of a transaction on the Ethereum system to construct an HMAC-based multiple-bit embedding scheme [46]. Frkat *et al.* presented a method for hidden botnet communication that exploits the digital signatures used in blockchains to inject covert messages [47].

We investigate these related works and sum up a general framework. Figure 1 shows the general framework of the blockchain-based covert communication model. It should be pointed out that the sender $\mathcal{S}$ and receiver $\mathcal{R}$ need to agree on something ahead, and the covert-message-embedded transaction is indistinguishable from the general transaction.

Finally, we study schemes similar to our research line and carry out a detailed comparison in terms of confidentiality of messages, concealment of communication behaviors, anonymity of communication entities, computational costs, communication costs, cryptocurrency costs, key management issues, and ciphertext leakage issues, as shown in Table 1.

3 Preliminaries

3.1 Notation

For any string s_1, s_2, $|s_1|$ denotes the length of s_1, $s_1||s_2$ denotes their concatenation. For any $i \in \mathbb{N}^+$, $[i]$ denotes integer set$\{1, 2, \ldots, i\}$. For any $i, j \in \mathbb{N}$ with $i < j$, $[i, j]$ denotes integer set $\{i, i+1, \ldots, j\}$. For any non-empty set $\mathcal{X}, x \xleftarrow{\$} \mathcal{X}$ denotes sampling uniformly x from $\mathcal{X}$. For any randomized algorithm $Alg(x), y \xleftarrow{\$} Alg(x)$ denotes the random output of $Alg(x)$. For any deterministic algorithm $Alg(x), y =$

$Alg(x)$ denotes the deterministic output of $Alg(x)$. For n elements $a_1, a_2, \ldots, a_n$, we denotes the set $\{a_i\}_{i\in[n]}$ as A.

3.2 Basic theory

(1) **Public-Key Encryption.** A public-key encryption scheme PKE consists of the following algorithms:

(a) $Setup(1^n)$ takes as input 1^n and returns the public parameter pp.
(b) $Gen(pp)$ takes as input pp and returns a public/secret key pair (pk, sk).
(c) $Enc(pk, m)$ takes as input pk and a plaintext m, and returns a ciphertext ct.
(d) $Dec(sk, ct)$ takes as input sk and ct, and returns m' or an abort symbol $\perp$.

Correctness. PKE is correct if, let $\mathcal{M}$ be the plaintext space, for any $m \in \mathcal{M}$,

$$\Pr\big[Dec(sk, ct) \neq m : pp \xleftarrow{\$} Setup(1^n); (pk, sk) \xleftarrow{\$} Gen(pp); ct \xleftarrow{\$} Enc(pk, m)\big] \leq negl(n).$$

Security. PKE is CPA secure for any probabilistic polynomial time (PPT) adversary $\mathcal{A}_1$ and $\mathcal{A}_2$,

$$|\Pr[b = b' : (m_0, m_1, st) \xleftarrow{\$} \mathcal{A}_1(pp, pk); b \xleftarrow{\$} \{0,1\}; ct' \xleftarrow{\$} Enc(pk, m_b); b' \xleftarrow{\$} \mathcal{A}_2(st, ct')] - \frac{1}{2}| \leq negl(n).$$

(2) **Entropy Smoothing Hash Functions.** Let $\mathcal{H} = \{H_k\}_{k\in\mathcal{K}}$ be a keyed hash function family associated with key space $\mathcal{K}$, groups X, Y, and hash function $H_k : X \to Y$. We say $\mathcal{H}$ is entropy smoothing for any PPT adversary $\mathcal{A}$, and $k \xleftarrow{\$} \mathcal{K}$, and $x, x' \xleftarrow{\$} X$,

$$\Pr[\mathcal{A}(k) \to (x \neq x') \wedge H_k(x) = H_k(x')] \leq negl(n),$$

$$|\Pr[\mathcal{A}(k, H_k(x)) = 1 | x \xleftarrow{\$} X] - \Pr[\mathcal{A}(k, y) = 1 | y \xleftarrow{\$} Y]| \leq negl(n).$$

(3) **Blockchain.** Blockchain [49] technology represents a secure and trusted decentralized distributed ledger, maintained by a network of interconnected nodes. This infrastructure [50, 51], devoid of a central authority, ensures that the blockchain possesses inherent security features such as immutability and unforgeability. Each node within this network maintains an identical copy of the ledger, chronicling transactions from their inception to the most recent ones. This ensures that once a transaction is recorded on the blockchain, it becomes immutable, preventing any unauthorized tampering or alteration.

When a new transaction enters the blockchain, the responsible node performs a rigorous verification process. This involves checking the digital signature of the transaction to ensure it meets the predefined criteria and standards. Once the verification is successful, the node proceeds to broadcast the transaction to all other nodes in the network. Each node then validates the transaction independently, accepting it as legitimate and adding it to their respective ledgers. This collective validation ensures the integrity and authenticity of each transaction recorded on the blockchain, fostering trust and transparency among all participants [52–54].

3.3 System and adversary model

System model. In BSAGINs, each entity that is distributed in the domains of space, air, and ground communicates with each other *via* the blockchain. Figure 2 depicts the system model. In CC-BSAGINs, there are two entities: the sender $\mathcal{S}$ and the receiver $\mathcal{R}$. $\mathcal{S}$ and $\mathcal{R}$ operate within distinct domains, while the blockchain refers to a blockchain integrated into the SAGINs. $\mathcal{S}$ can send overt and covert messages to $\mathcal{R}$. Overt message transmission between $\mathcal{S}$ and $\mathcal{R}$ is facilitated through a public channel established on the blockchain. At the same time, two communication entities through which the public message flows can carry out covert communication through the covert channel. Thus, the covert message is sent.

Adversary model. In CC-BSAGINs, there are two main types of adversaries: honest but curious receivers and network eavesdroppers.

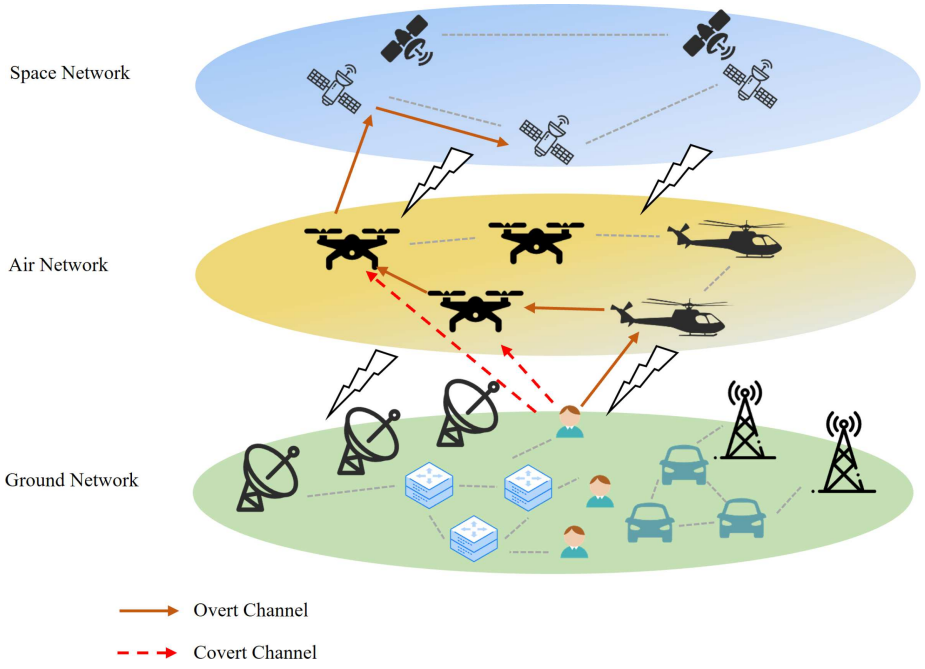

Figure 2. System model

(1) **Honest but curious receiver.** She or he attempts to know the identity of those utilizing the identical covert anonymous communication protocol or speculates on the sender's identity. In comparison to network eavesdroppers, honest but curious receivers can receive the covert message, thereby affording them greater advantages in detecting. Such an adversary is an inside adversary who knows the receiver's public key. She or he wants to know about the other pair of covert communicators while having covert communication with the sender. Furthermore, she or he wants to know the real identity of the sender with whom he is engaged in covert communication. She or he can interact with $\mathcal{S}$.

(2) **Network eavesdropper.** The network eavesdropper detects covert channels through the act of intercepting and analyzing network traffic. Given the numerous occurrences of network eavesdropping incidents, it is reasonable to assume that the eavesdropper has robust monitoring capabilities with respect to end-to-end data transmission. Since the blockchain network is public, and blockchain data is permanently stored, the network eavesdropper has enough time to detect and analyze all transaction data. Any noticeable differences will reveal covert channels. Such an adversary is an external adversary who only knows that someone on the blockchain-integrated SAGINs is conducting covert communication. She or he also cannot interact with $\mathcal{S}$ and $\mathcal{R}$.

3.4 Design goals

We propose a two-step paradigm of covert communication, where the ciphertext of covertly sent data would not appear in the network, and the receiver can extract the data from a secure transformation mechanism. Then we integrate the above mechanism into the communication scheme and design a system, dubbed CC-BSAGINs. The design goals are summarized as follows.

(1) **No key management issues.** In the realm of SAGINs, the volatile and dynamic nature of the environment poses unique challenges to secure communication. One such challenge is the establishment of secure key agreements, and exchanges among the various equipment and devices deployed within these networks. Given the high risk of equipment loss or compromise, traditional methods of key agreements can often become impractical or unfeasible. However, our solution, CC-BSAGINs, offers an approach to this problem. CC-BSAGINs stands out by enabling covert communication without the need for prior key agreements. This paradigm shift eliminates the dependency on complex and potentially vulnerable key exchange mechanisms, thus greatly simplifying the communication process.

The core advantage of CC-BSAGINs lies in their ability to guarantee the concealment of communication behavior, even in the absence of a pre-established key. This means that equipment and devices within SAGINs can clandestinely transmit sensitive information or instructions without attracting undue attention or inviting security breaches. Such communication remains undetectable and untraceable. CC-BSAGINs significantly enhance the security and reliability of communication in SAGINs. It not only mitigates the risks associated with equipment loss or compromise but also reduces the complexity and overhead involved in traditional key management processes. As a result, CC-BSAGINs stand as a robust and efficient solution for secure communication in the dynamic and challenging environment of SAGINs.

(2) **No ciphertext leakage.** In CC-BSAGINs, the ciphertext is not directly stored on the blockchain. $\mathcal{S}$ matches the ciphertext with the transaction and then sends the transaction index to $\mathcal{R}$ through a covert channel. After receiving the index, $\mathcal{R}$ extracts the transaction and obtains the ciphertext from the blockchain. Therefore, the ciphertext is not leaked on the blockchain because we transfer the treasure map of the ciphertext rather than the ciphertext. However, some works [30, 41, 44, 46, 48] store the ciphertext on the INPUT field, signature, address, and so on. In this way, the ciphertext is permanently stored on the blockchain because of the blockchain's immutable and distributed nature, which makes the ciphertext available to anyone. Although the current encryption algorithms are computational security, with the continuous progress of mathematical theory and computing technology, the existing encryption algorithms have the risk of being compromised. Once compromised, the corresponding plaintext of the ciphertext can be directly recovered by the adversary. This is the risk of ciphertext leakage.

(3) **Compatible with existing public blockchain.** Compatibility with existing public blockchain is crucial for the success of covert communication schemes. The reason for this lies in the fundamental nature of blockchain networks: the more normal transactions occur within the blockchain, the more effectively it camouflages communication behavior. This makes the most popular public blockchain ideal candidates for covert communication, as they boast a high volume of transactions and a widespread user base. However, to integrate covert communication into this popular public blockchain, it is essential that the communication scheme is fully compatible with the existing blockchain systems. This means that the scheme should operate seamlessly without the need to modify the core protocols of these blockchain systems. Any modifications to the underlying blockchain protocols could potentially introduce vulnerabilities or disrupt the integrity of the network, which is unacceptable. Therefore, the design of a covert communication scheme must take into account the specific characteristics and limitations of the target public blockchain. It should leverage the existing functionalities and mechanisms of the blockchain to achieve its objectives while adhering to the principles of compatibility and non-intrusive integration. By doing so, we can ensure that the covert communication scheme remains undetectable within the normal transactions of the blockchain, maintaining the security and integrity of both the communication and the blockchain network itself.

4 Proposed CC-BSAGINs

In this section, we introduce the CC-BSAGINs which frees the sender from maintaining long-term cryptographic secret keys and ensures data confidentiality even if the sender is controlled by the adversary. In the face of the highly complex and adversarial network environment of the SAGINs, as one of the nodes, unmanned aerial vehicle (UAV) has the risk of being controlled by adversaries. To defend against such adversaries, we consider the strongest assumption. In this application scenario, nodes in SAGINs can send messages in a covert and secure way by CC-BSAGINs. CC-BSAGINs consists of five algorithms **Setup**, **TxRandom**, **CovertchannelSend**, **TxFind**, and **TxDec**. Figure 3 shows the sketch of CC-BSAGINs. Then we instantiate it.

4.1 Paradigm

These five algorithms are listed below. Figure 4 shows all the algorithms of the scheme in detail.

(1) **Setup**(ℓ) takes as input a security parameter ℓ and returns public parameters $\{\mathsf{PKE}, H_k, Add\}$, where PKE is a public-key encryption, H_k is an entropy smoothing hash function, and Add is an

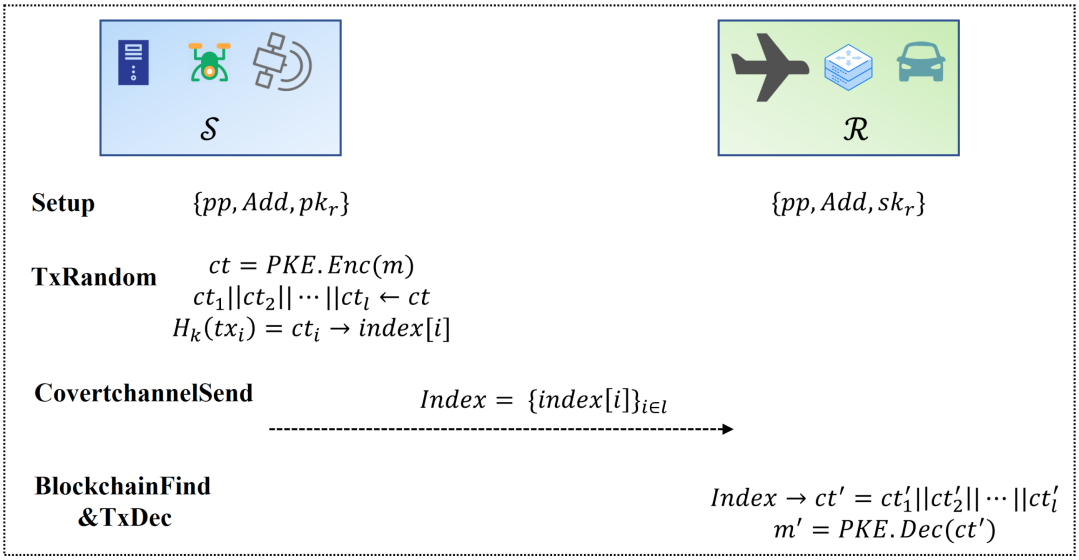

Figure 3. Sketch of CC-BSAGINs

account address on a blockchain. In this algorithm, an encryption, a hash function, and a blockchain instantiation are determined.

(2) **TxRandom**(pk, dm) takes as input a public key pk of $\mathcal{R}$ and a plaintext message m and returns transaction index set $Index$. In this algorithm, $\mathcal{S}$ encrypts a message m with $\mathcal{R}$'s public key. Then the ciphertext is divided into l slices according to the length of j, and the transaction whose hash value is equal to the slice is found on the blockchain.
   (a) $\mathcal{S}$ encrypts a message m with $\mathcal{R}$'s public key and gets ciphertext ct.
   (b) $\mathcal{S}$ divides the ciphertext into l slices of length j, that is $ct = ct_1||ct_2||\cdots||ct_l$ $(l \cdot j = |ct|, |ct_l| = j)$.
   (c) $\mathcal{S}$ finds a transaction tx_i on the amount address Add whose hash value is equal to the ciphertext slice, that is $ct_i = H_k(tx_i)$, and records the index value of the transaction $index[i]$. That is, ciphertext slices are matched with transactions one by one.
   (d) Finally, $\mathcal{S}$ gets the set of index $Index$. $(Index = \{index[i]\}_{i\in[l]})$.

(3) **CovertchannelSend**$(message)$ implies sending a message through a covert channel. In this algorithm, $\mathcal{S}$ sends the set of index $Index$ to $\mathcal{R}$ through a covert channel.

(4) **BlockchainFind**$(Index)$ takes as input a transaction index set $Index$ and returns the transaction set Tx. In this algorithm, after receiving the index value set $Index$, $\mathcal{R}$ finds the transaction corresponding to the index value on the amount address Add and finally extracts the transaction set Tx. $(Tx = \{tx_i\}_{i\in[l]})$.

(5) **TxDec**(sk, Tx) takes as input a secret key sk of $\mathcal{R}$ and a transaction set Tx and returns m'. In this algorithm, $\mathcal{R}$ recovers the ciphertext by computing the hash value of the transaction and finally decrypts the ciphertext into plaintext with her or his private key sk.
   (a) $\mathcal{R}$ computes the hash value of each transaction to obtain the ciphertext slices, that is $ct'_i = H_k(tx_i)(i \in [l])$.
   (b) $\mathcal{R}$ concatenates the ciphertext slices to obtain the ciphertext, that is $ct' = ct'_1||ct'_2||\cdots||ct'_l$.
   (c) $\mathcal{R}$ decrypts the ciphertext ct' with private key sk. Ultimately, $\mathcal{R}$ gets the plaintext m'.

4.2 Construction of the CC-BSAGINs

We construct an efficient instantiation, where the PKE is based on ElGamal encryption, and the covert channel is based on [41].

(1) **Setup.** With the security parameter ℓ, the public parameters $\{p, G, g, H, Enc(\cdot), Dec(\cdot), Add\}$ are determined, where G is a multiplicative group with prime order p, g is a generator of G, H:$\{0,1\}^* \rightarrow Z_p^*$ is entropy smoothing hash functions, $Enc(\cdot)$ is ElGamal encryption algorithm, and $Dec(\cdot)$ is ElGamal

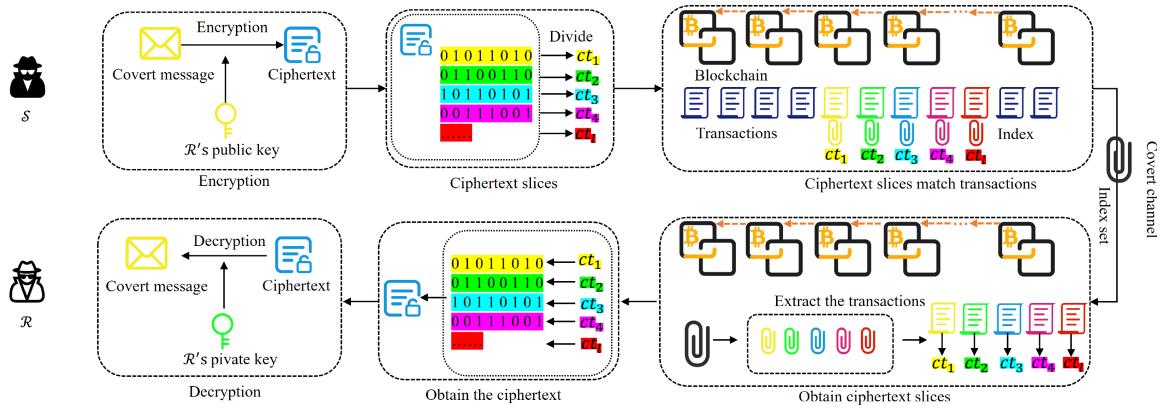

Figure 4. The workflow of CC-BSAGINs

decryption algorithm, *Add* is an account address on the blockchain. $\mathcal{R}$ uniformly chooses an element a in the group G as the private key, then computes g^a as the public key. Thus, $\mathcal{R}$'s key pair is $(pk, sk) = (g^a, a)$

(2) **TxRandom.** $\mathcal{S}$ encrypts a message m with $\mathcal{R}$'s ElGamal public key. The ciphertext is divided into l slices, and each slice is 8-bit. Then $\mathcal{S}$ finds a transaction on the blockchain amount address *Add* whose hash value is equal to the slice and records the index of the transaction. Thus, $\mathcal{R}$ gets the set of index $Index$.

(a) $\mathcal{S}$ uniformly chooses $r \xleftarrow{\$} Z_p$ and computes $ct = (g^r, (g^a)^r \cdot m)$. ct is a pair of ciphertext corresponding to the plaintext m.
(b) $\mathcal{S}$ divides the ct into l slices of length 8 bits, $ct = ct_1||ct_2||\cdots||ct_l$.
(c) $\mathcal{S}$ computes $H(tx_i)$ and records $index[i]$ the index of transaction of the blockchain address *Add* for which $H(tx_i) = ct_i$. Finally, $\mathcal{S}$ obtains the index set $Index$.

(3) **CovertchannelSend.** $\mathcal{S}$ sends the set of index $Index$ to $\mathcal{R}$ through a covert channel based on [41].

(4) **BlockchainFind.** $\mathcal{R}$ finds the transaction corresponding to the index on the address *Add*. Ultimately, $\mathcal{R}$ obtains the transaction set Tx. Namely, $\mathcal{R}$ extracts the transactions according to the $Index$.

(5) **TxDec.** $\mathcal{R}$ recovers the ciphertext slices and decrypts the ciphertext into plaintext with her or his ElGamal private key sk.

(a) $\mathcal{R}$ computes $ct'_i = H(tx_i)$ and obtain the ciphertext $ct' = (C_1, C_2)$.
(b) $\mathcal{R}$ computes $m' = C_2/(C_1)^a$. In the end, $\mathcal{R}$ gets plaintext m'.

4.3 Advantages of CC-BSAGINs

We compare CC-BSAGINs with other works [30, 41, 44], highlighting its unique advantages in the following aspects: no key management issues, no ciphertext leakage, and low cost (especially in terms of cryptocurrency consumption). Table 2 shows the advantages of CC-BSAGINs.

(1) **No key management issues.** Most of the covert communication systems are built using blockchain need key agreements between $\mathcal{S}$ and $\mathcal{R}$, such as the private key of the blockchain account, and the secret key of symmetric encryption. However, in SAGINs, due to the easy loss, damage, and capture of the device, and the complexity and fragility of the network, it is impractical to carry out key agreements between the two parties. For example, it is obviously not practical for satellites in space, charging stations on the side of the road, drones in the sky, and TV towers in the suburbs to exchange keys. Gao *et al.* proposed a kleptography-based covert data transmission mechanism [41], and Hartl *et al.* proposed a covert channel scheme in EdDSA [30], the sender and receiver need key agreement so that they can share a private key. Furthermore, Zhang *et al.* proposed a covert communication scheme [44],

Table 2. Comparison

Scheme	No key management issue	No ciphertext leakage	Low costs
[41]	Private key	INPUT field	NO
[30]	Private key	Signature	YES
[44]	Private key and secret key	NO	NO
CC-BSAGINs	NO	NO	YES

in which the sender and receiver not only share a private key of blockchain but also agree on a secret key for threshold secret sharing [55].

(2) **No ciphertext leakage.** The direct storage of ciphertext on the blockchain poses a risk due to its immutable and distributed nature, as advancements in mathematical theory and computing technology, particularly the advent of quantum algorithms and computers, have rendered current mainstream encryption algorithms vulnerable. Once these vulnerabilities are exploited, confidential data stored on the blockchain will be exposed, rendering this situation unacceptable. To be specific, Gao *et al.* proposed the kleptography-based covert data transmission mechanism that [41] stores the ciphertext in the INPUT field of the blockchain, and Hartl *et al.* proposed the covert channel in EdDSA [30] that stores the ciphertext in the random number in signature. However, in CC-BSAGINs, the treasure map of ciphertext rather than ciphertext itself is stored on the blockchain.

(3) **Low costs.** The implementation of a blockchain-based covert communication system typically involves utilizing transactions as the transmission medium for concealed information, often by embedding such information within digital signatures and INPUT fields. However, conducting transactions on the blockchain necessitates fuel in the form of cryptocurrency, thereby resulting in high monetary costs associated with this type of scheme. Specifically, in proposed the kleptography-based covert data transmission mechanism [41], $\mathcal{S}$ costs cryptocurrency to send a transaction on the blockchain so that can embed the covert information in the INPUT fields.

5 Security analysis

We follow the security definitions in [56, 57]. CC-BSAGINs differs from ℓ-sender-anamorphic encryption (ℓ-sender AME) in [56] in only a few ways: we use the transaction index to replace anamorphic ciphertext. Thus, the security proof of our proposed scheme is based on the proof of ℓ-sender-AME in [56]. We analyze the security of CC-BSAGINs from three aspects.

Since CC-BSAGINs has the advantage of no ciphertext leakage issue, to investigate the security-enhanced extent, we add the security-enhanced analysis compared with a scheme [41] that does not have the advantage.

5.1 Confidentiality of messages

Theorem 1. *If H_k is modeled as a random oracle H, and PKE is CPA-secure, then CC-BSAGINs is CPA-secure.*

Proof. Let H_1 denote the game for $\mathcal{A}$ in ℓ-sender-anamorphic encryption in [56]. Game H_2 is the same as H_1 except that all the ciphertexts in ct are hash values sampled from $H_k(\{0,1\}^l)$ uniformly, instead of generated by encrypting the plaintext. Game H_3 is the same as H_2 except that the (FPK, FSK) are not generated. Games H_1, H_2, and H_3 are shown below. Since H_1 has been proven to be CPA-secure, to demonstrate that H_2 is also CPA-secure, it suffices to show that the PPT-adversary $\mathcal{A}$ cannot distinguish between a hash value and a ciphertext encrypted from a public key with a significant advantage. Therefore,

$$\Pr[\mathsf{H}_1(n) = 1] \leq \frac{1}{2} + negl(n),$$

$$\Pr[\Pi_{\mathcal{A}}^{Oracle}(n) = 1] = \frac{1}{2}.$$

Then we have

$$\Pr[\mathsf{H}_1(n)=1]-\Pr[\Pi_{\mathcal{A}}^{Oracle}(n)=1]|\leq negl(n).$$

Assuming that there is a PPT adversary $\mathcal{A}$ who can distinguish from a significant probability the output returned by the oracle, then there must be a PPT adversary $\mathcal{B}$ who can win H_2 with a significant probability. However, there is no such PPT adversary $\mathcal{A}$, so there is no such PPT adversary $\mathcal{B}$. Namely,

$$|\Pr[\mathsf{H}_2(n)=1]-\Pr[\Pi_{\mathcal{A}}^{Oracle(n)}=1]|\leq negl(n).$$

In the same way, there is no such PPT adversary $\mathcal{C}$ who can win H_3 with a significant probability. We have $|\Pr[\mathsf{H}_3(n)=1]-\Pr[\mathsf{H}_2(n)=1]|\leq negl(n)$, then

$$|\Pr[\mathsf{H}_3(n)]=1|\leq negl(n).$$

Game H_3 corresponds to CC-BSAGINs, thus CC-BSAGINs is CPA-secure. This concludes the proof. ■

$\mathsf{H}_1(1^n)$	$\mathsf{H}_2(1^n)$	$\mathsf{H}_3(1^n)$
$pp\xleftarrow{\$}Setup(1^n)$	$pp\xleftarrow{\$}Setup(1^n)$	$pp\xleftarrow{\$}Setup(1^n)$
$(fpk_i,fsk_i)_{i\in[l]},(dpk,dsk)\xleftarrow{\$}Gen(pp)$	$(fpk_i,fsk_i)_{i\in[l]},(dpk,dsk)\xleftarrow{\$}Gen(pp)$	$(dpk,dsk)\xleftarrow{\$}Gen(pp)$
$b\xleftarrow{\$}\mathcal{A}^{ENC_1}(pp,FPK)$	$b\xleftarrow{\$}\mathcal{A}^{ENC_2}(pp,FPK)$	$b\xleftarrow{\$}\mathcal{A}^{ENC_3}(pp,FPK)$
return b	return b	return b

$\mathsf{ENC}_1(1^n)$	$\mathsf{ENC}_2(1^n)$	$\mathsf{ENC}_3(1^n)$
$R\xleftarrow{\$}fRandom(FPK,FM,dpk,dm)$	$Index\xleftarrow{\$}Tx.Random(dpk,dm)$	$Index\xleftarrow{\$}Tx.Random(dpk,dm)$
return $\{ct_i\}_{i\in[l]}$	return $\{ct_i'\}_{i\in[l]}$	return $\{ct_i'\}_{i\in[l]}$

5.2 Anonymity of communicating entities

It is well known that when a blockchain user wants to create a blockchain account, she/he does not need any identity information of herself/himself, only a string of fixed size as her/his private key. The identity of the blockchain a determined by the blockchain address, which is the hash of the public key. Thus, blockchain is an anonymous system. Therefore, the communication entities in the communication system based on blockchain also have anonymity. Despite the blockchain account being a pseudonym generated by the user that is not directly related to his or her real identity, current research shows that through heuristic analysis of transaction records, the clustering relationship of pseudonymous address can be deduced, and even the user's real identity can be inferred [58]. Hence, the anonymity of communicating entities cannot be fully guaranteed by only using blockchain technology. The proposed solution in CC-BSAGINs is as follows: communicating entities do not require blockchain accounts; rather, they only need to accomplish communication by observing transactions associated with a specific account on the blockchain. Furthermore, when communication entities need to transmit messages, we employ covert channels.

Definition 1. A scheme is anonymous for communicating entities if for any PPT adversary $\mathcal{A}$, the tokens of communication entities are indistinguishable from the uniform string.

Theorem 1. *If $Hash$ is modeled as a random oracle H, then CC-BSAGINs are anonymous for communicating entities.*

Proof. In blockchain-integrated SAGINs, every communication entity can construct an account on the blockchain without any real private information. Furthermore, the address is the identity of the blockchain account, where $address = Hash(account.publickey)$. Because the tokens of the communication entities are the hash values, they are indistinguishable from the uniform string for any PPT adversary $\mathcal{A}$. This concludes the proof. ■

5.3 Concealment of communication behaviors

Inspired by [57, 59], we define the communication behavior concealment.

Definition 2. A scheme is covert for communication behaviors if for any PPT adversary $\mathcal{A}$, a covert message/ciphertext is indistinguishable from an overt message/ciphertext.

Theorem 2. *If PKE is a CPA-secure encryption and H_k is modeled as a random oracle H, then CC-BSAGINs is covert for communication behaviors.*

Proof. If $\mathcal{A}$ succeeds in breaking *communication behavior concealment* in CC-BSAGINs with a non-negligible probability, we can construct an efficient $\mathcal{A}'$ to break a CPA-secure PKE with a non-negligible probability. Specifically, $\mathcal{A}'$ uses $\mathcal{A}$ as a subroutine and can break the PKE as follows. Oracle $\mathcal{I}$ chooses a secret $c \xleftarrow{\$} Z_p$ and computes $H_k(c)$ and $\mathsf{PKE}.Enc(c)$. After $\mathcal{A}'$ obtains $H_k(c), \mathsf{PKE}.Enc(c)$ from $\mathcal{I}$, $\mathcal{A}'$ queries $\mathcal{A}$ with $(H_k(c), \mathsf{PKE}.Enc(c))$. Upon receiving $(H_k(c), \mathsf{PKE}.Enc(c))$, $\mathcal{A}$ distinguishes between the $H_k(c)$ and $\mathsf{PKE}.Enc(c)$ with a non-negligible probability. However, there is no such adversary that can break a CPA-secure PKE with a non-negligible probability. Therefore, CC-BSAGINs is covert for communication behaviors. This concludes the proof. ■

In a word, communication behavior concealment means that for any PPT adversary $\mathcal{A}$, general ciphertexts (overt messages) and special ciphertexts (covert messages) cannot be distinguished with a significant advantage.

5.4 Security enhanced analysis

Let us recall the kleptography-based scheme in [41]. The scheme consists of two algorithms **Special transaction creation** and **Special transaction filtering**. The sketch of it is listed below.

(1) **Special transaction creation**(m, pk_r) takes as input the receiver's public key pk_r and a plaintext message m, and returns a general transaction T_n and a special (covert-message-embedded) transaction T_s.

(2) **Special transaction filtering**($TX = \{T_0, ..., T_n\}, (pk_r, sk_r)$) takes as input the transaction set TX, the receiver's public key pk_r, and the receiver's private key sk_r, and returns a special transaction set TX_s and a private key extracted from a special transaction set SK_s.

Algorithm 1 and Algorithm 2 show the detail.

Theorem 3. *If ECC is a CPA-secure encryption, then the kleptography-based scheme is CPA-secure.*

Proof. If $\mathcal{A}$ succeeds in breaking *CPA* in the kleptography-based scheme with a non-negligible probability, we can construct an efficient $\mathcal{A}'$ to break a CPA-secure ECC with a non-negligible probability. However, there is no such $\mathcal{A}'$. Therefore, there is no such $\mathcal{A}$. This concludes the proof. ■

Theorem 4. *If the ciphertext is stored on the INPUT field of a transaction directly, then the kleptography-based scheme is **not** covert for communication behaviors.*

Proof. Because it stores the ciphertext on the INPUT field of a transaction, the $\mathcal{D}$ can distinguish between the general transaction from the special transaction (covert-message-embedded transaction). Therefore, the kleptography-based scheme is **not** covert for communication behaviors. This concludes the proof. ■

This scheme is CPA secure. However, it has a fatal drawback: it stores the ciphertext directly on the INPUT field of a transaction, which makes the ciphertext available to anyone. In CC-BSAGINs, the ciphertext is generated by transactions on the blockchain as a "seed" because of $ct_i = H(tx_i)$. The ciphertext does not appear on the blockchain, the "seed" is stored on the blockchain.

6 Performance evaluation

We implement a prototype in Python 3.9 and conduct experiments to evaluate the performance with a security parameter of 1024 bits, and PKE is implemented using ElGamal encryption. The experiments

Algorithm 1: Special transaction creation.

Input: The receiver's public key: pk_r; The plaintext message: m.
Output: A general transaction T_n; A special (covert-message-embedded) transaction T_s.

set $(pk_s, sk_s) = ECC.KeyGen(\lambda)$;
set $addr = CreateAccount(pk_s)$;
set $e = ECC.Enc(m, pk_s)$;
set $data_{tn} = CreateTrans(addr, null, params_0)$;
set $data_{ts} = CreateTrans(addr, e, params_1)$;
set $K_1 \leftarrow \{0,1\}^{\lambda}$;
set $\sigma_n = ECDSA.Sign(data_{tn}, sk_s, K_1)$;
set $\sigma_s = ECDSA_{KLE}.Sign(data_{ts}, sk_s, K_1, pk_r)$;
set $T_n = (data_{tn}, \sigma_n)$;
set $T_s = (data_{ts}, \sigma_s)$;
return T_n, T_s.

Algorithm 2: Special transaction filtering.

Input: The transaction set TX; The receiver's public key pk_r; The receiver's private key sk_r.
Output: A special transaction set TX_s; A private key extracted from a special transaction set SK_s.

init $TX_s = \{\}, SK_s = \{\}$;
for $i = 0; i \leq n; i++$ **do**
  extract $addr_i$ from $T_{(n-1)}$;
  find last transaction $T^{prev}_{(n-1)}$ associated with input address $addr_i$;
  extract $\sigma_{(n-i)}$ from $T_{(n-i)}$;
  extract $\sigma^{prev}_{(n-i)}$ from $T_{(n-1)}$;
  set $sk_i = skExtract(T_{(n-i)}, T^{prev}_{n-i}, \sigma_{(n-i)}, \sigma^{prev}_{n-i}, sk_r, pk_r, pk_s)$;
  set $pk_i = ECC.generatePk(sk_i)$;
  if $pk_s \neq pk_i$ **then**
    i++;
    continue;
  else
    add $T_{(n-i)}$ to TX_s;
    add sk_i to SK_s;
    i++;
  end;
return TX_s, SK_s.

are conducted on a laptop with Windows 10, an AMD Ryzen 7 5800H with Radeon Graphics 3.2 GHz CPU, and 32 GB 3200 Mhz DDR4 of RAM. Ethereum is used as the underlying blockchain and Etherscan is used as the Application Programming Interface (API) function.

There are five algorithms in CC-BSAGINs, and we will analyze each algorithm one by one. First, **Setup** usually takes about 1 second with a security parameter of 1024 bits. Since $\mathcal{R}$ will not update the secret key frequently in a short period of time, **Setup** is not executed every time. Thus, its costs are very small. Then **TxRandom** includes the encryption and matching, therefore, it takes high. **CovertchannelSend** depends on the specific covert channel algorithm, we do not consider its costs here. **BlockchainFind** extracts the ciphertext from the transactions and its costs between **TxRandom** and **TxDec**. **TxDec** is a decryption process, which only consumes at the millisecond level.

Since blockchain technology is used, the costs of cryptocurrency on the blockchain should also be considered.

Furthermore, we compare computational, communication, and cryptocurrency costs of CC-BSAGINs with the kleptography-based scheme in [41], the Shamir threshold-based scheme in [44], the EdDSA-based subliminal channel in [30], the Hash-based multiple-bit embedding scheme in [46], and the Zcash-based subliminal channel in [48]. Table 1 and Figure 5 show the comparison.

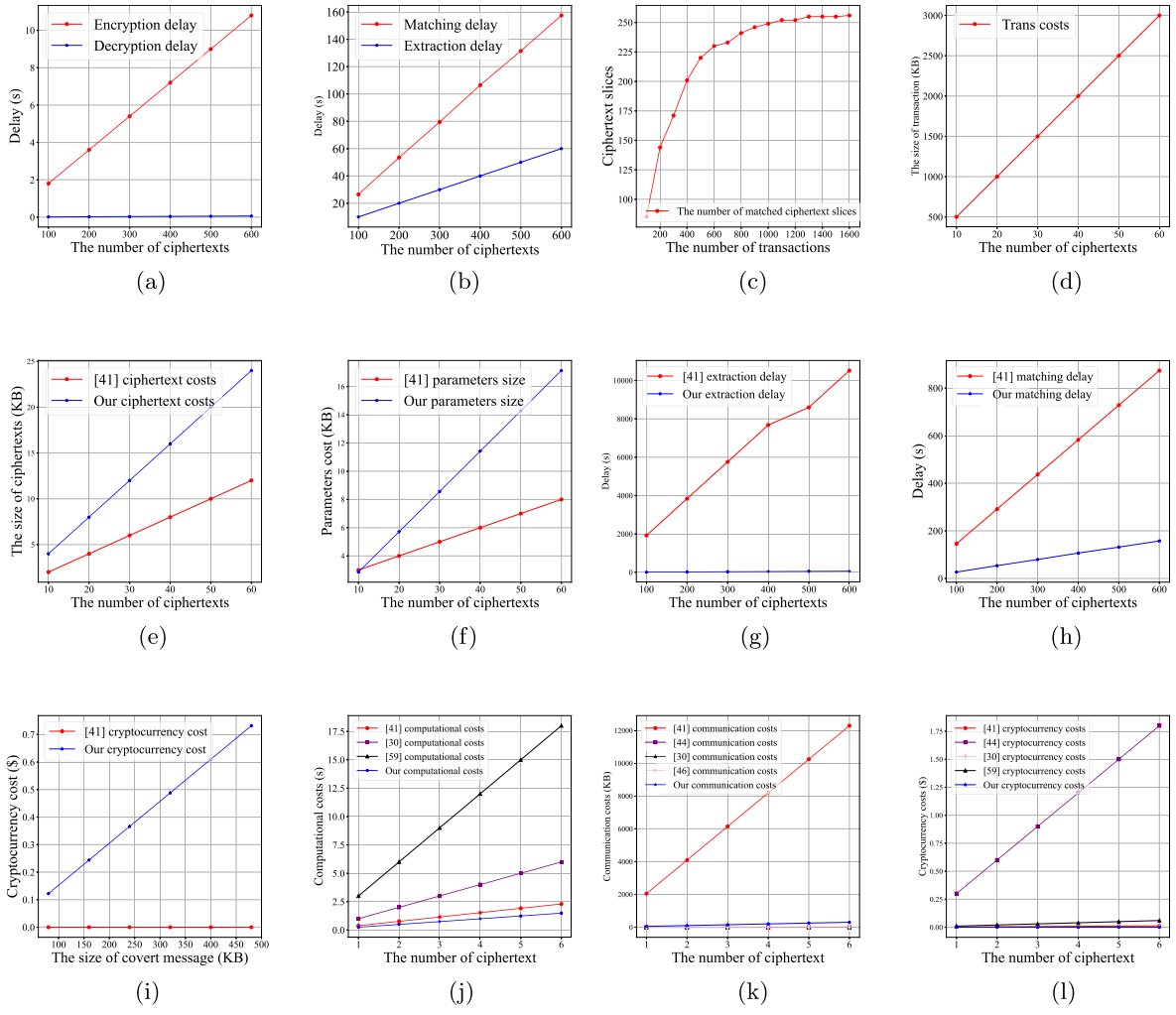

Figure 5. Costs. (a) Enc/Dec delay. (b) Match/Extract delay. (c) Matched slices. (d) Transactions costs. (e) Ciphertext costs. (f) Parameter size. (g) Extract delay. (h) Matching delay. (i) Cryptocurrency costs comparison. (j) Computational costs. (k) Communication costs. (l) Cryptocurrency costs.

6.1 Computational costs

We evaluate the computational costs in two aspects.

Sender. The delay of $\mathcal{S}$ is mainly divided into two parts: encryption delay and matching delay. It should be pointed out that because $\mathcal{S}$ needs to connect to Ethereum, the delay in connecting to Ethereum is related to Ethereum service providers and network connectivity. Since this is not the focus of this paper, we ignore this part of the delay. With a security parameter of 1024 bits, the encryption delay is usually in the order of milliseconds. However, the matching delay is usually in the order of seconds. Furthermore, the ciphertext is divided into 256 slices according to the length of 8 bits. Figure 5a shows the encryption delay of the sender, Figure 5b shows the matching delay of the sender, and the independent variable is the number of ciphertexts. Figures 5a and 5b show us that the relationship between the number of ciphertexts and the delay is linear. The simulation test shows that it takes about 1600 transactions to match the 256 slices completely. Figure 5c shows the relationship between the number of transactions and matched ciphertext slices. Clearly, they are logarithmic. $\mathcal{S}$ gets the ciphertext of size 2048 bits, and it takes 1600 transactions to match. Though a transaction costs about 1 MB, we only need the transaction index to cost about 256 bits. Thus, the 1600 transactions of indexes are about 50 KB. Figure 5d shows the relationship between the number of ciphertexts and transaction index costs. Finally, a 2048-bit ciphertext needs 50 KB transaction indexes based on CC-BSAGINs.

Table 3. Computational costs

Encryption delay	Decryption delay	Matching delay	Extract delay
0.018 s	0.0001 s	0.2647 s	0.1 s

Table 4. Communication costs

Public parameter size	Ciphertext size	Transactions indexes size
2304 bits	2048 bits	50 KB

Receiver. As mentioned earlier, we also ignore this part of the delay for the receiver to connect to Ethereum. The delay of $\mathcal{R}$ is mainly divided into two parts: decryption delay and extraction delay. Figure 5a and 5b show the decryption and extraction delay. Compared to $\mathcal{S}$, the delay of decryption and extraction is much lower.

Comparison. Figure 5g shows the relationship between the number of ciphertexts and extracting delay. It can be seen that for every 100 ciphertexts, our extraction delay is 10 seconds, and that of the scheme [41] is 1920 seconds. Figure 5h shows the relationship between the number of ciphertexts and matching delay. It can be seen that for every 100 ciphertexts, our matching delay is 26.47 s, and that of the scheme [41] is 145.6 s. Thus, CC-BSAGINs are much lower than [41] in the computational costs. Figure 5j shows the computational cost comparison of CC-BSAGINs with other schemes [30, 41, 48].

Table 3 shows the computational costs corresponding to each ciphertext in detail.

6.2 Communication costs

We evaluate the communication costs in two aspects.

Sender. For instantiation of CC-BSAGINs, we use ElGamal encryption as the building block of PKE with a security parameter of 1024 bits and Ethereum as the building block of the blockchain. Thus the public parameters are $\{p, G, g, H, Add\}$. The prime order p and the generator g determine the multiplicative group G. Add in the address of Ethereum and is 256 bits in size. Therefore, the size of public parameters is 2304 bits. Figure 5f shows the relationship between the number of ciphertexts and the size of public parameters, and comparison with [41].

Receiver. $\mathcal{R}$ obtains the about transactions indexes of size 50 KB from the covert channel and extracts the ciphertext of size 2048 bits, namely 256 Bytes. Figure 5e shows the relationship between the number of ciphertexts and ciphertexts costs, and the comparison with [41].

$\mathcal{R}$ obtains transaction indexes of approximately 50 KB in size from the covert channel and extracts ciphertexts sized at 2048 bits, equivalent to 256 Bytes. The relationship depicted in Figure 5e demonstrates how the number of ciphertexts impacts the overall cost.

Table 4 shows the communication costs corresponding to each ciphertext in detail. Figure 5k shows the communication costs comparison of CC-BSAGINs with other schemes [30, 41, 44, 46].

6.3 Cryptocurrency costs

The application of blockchain usually requires cryptocurrency, and the relationship between blockchain and cryptocurrency is similar to the relationship between car and fuel oil. Cryptocurrency is required to perform transactions on the blockchain, invoke smart contracts, and so on, but not for every operation, such as viewing transactions on the blockchain, cryptocurrency is not required. As the proposed scheme in this paper, we match the existing transactions on the blockchain with the ciphertext. It is essentially a lookup process and does not require the use of cryptocurrency. However, the proposed scheme in [41] must use cryptocurrency, because their scheme needs to send transactions on the blockchain. Furthermore, a lot of covert communication schemes based on blockchain require sending transactions. The costs of

cryptocurrency are very high, and the channel capacity of the covert communication scheme constructed by the blockchain is measured in bits. Therefore, the overhead of cryptocurrency is very large using this kind of covert communication scheme. For example, the proposed scheme in [41] requires approximately $ 0.122 in cryptocurrency per 80 Bytes. But CC-BSAGINs do not require cryptocurrency and cost $ 0 per 80 Bytes. Figure 5i shows the relationship between cryptocurrency costs and covert data size of the scheme [41] and CC-BSAGINs. Figure 5l shows the cryptocurrency costs comparison of CC-BSAGINs with other schemes [41, 44, 48].

7 Conclusion

In this paper, we have proposed a two-step paradigm of covert communication, where the ciphertext of covertly sent data would not appear in the network and the receiver can extract the ciphertext from a secure transformation mechanism. We also have instantiated the transformation using blockchain and an efficient index algorithm. Furthermore, we have integrated the above mechanism into a covert communication scheme and developed a system, in which we have formally proven the security and conducted a comprehensive performance evaluation.

For future work, we will investigate how to further reduce the computational and communication costs introduced by deploying CC-BSAGINs, since the devices in SAGINs have limited computation and network resources. The covert communications between the devices should be conducted as efficiently as possible. We will research on how to design a more efficient instantiation while achieving the same security guarantee as CC-BSAGINs.

Conflict of interest

The authors declare that they have no conflict of interest.

Data Availability

No data are associated with this article.

Authors' Contributions

Weijia Li and Yuan Zhang designed and coordinated the research program; Weijia Li, Yuan Zhang, and Xinyu He set up the methodology; Xinyu He and Yaqing Song performed the analyses; and Weijia Li and Yuan Zhang wrote the manuscript.

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions.

Funding

This work was supported in part by the National Key R&D Program of China under Grant 2023YFB3106500; in part by the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology (CAST) under Grant 2022QNRC001; in part by the Sichuan Science and Technology Program under Grant 2022ZDZX0038 and Grant 2023ZYD0142.

References

[1] Shang B, Yi Y and Liu L. Computing over space-air-ground integrated networks: Challenges and opportunities. IEEE Network 2021; **35**: 302–309
[2] Bao Z, Luo M, Wang H, et al. Blockchain-based secure communication for space information networks. IEEE Network 2021; **35**: 50–57.
[3] Ali M, Nelson J, Shea R, et al. Blockstack: A global naming and storage system secured by blockchains. In: Proc. USENIX ATC, 2016, 181–194.
[4] Tomescu A and Devadas S. Catena: Efficient non-equivocation via bitcoin. In: Proc. IEEE S & P, 2017, 393–409.
[5] Yang N, Guo D, Jiao Y, et al. Lightweight blockchain-based secure spectrum sharing in space-air-ground integrated iot network. IEEE Internet Things J 2023; **10**: 20 511–20 527.
[6] Liu X, Yang A, Huang C, et al. Decentralized anonymous authentication with fair billing for space-ground integrated networks. IEEE Trans Veh Technol 2021; **70**: 7764–7777.
[7] Huang C, Xue L, Liu D, et al. Blockchain-assisted transparent cross-domain authorization and authentication for smart city. IEEE Internet Things J 2022; **9**: 17 194–17 209.
[8] Wang D, Qi P, Zhao Y, et al. Covert wireless communication with noise uncertainty in space-air-ground integrated vehicular networks. IEEE Trans Intell Transp Syst 2021; **23**: 2784–2797.
[9] Chen X, Chang Z, Tang J, et al. Uav-aided multi-antenna covert communication against multiple wardens. In: Proc. IEEE ICC, 2021, 1–6.

[10] Luo X, Zhang P, Zhang M, et al. A novel covert communication method based on bitcoin transaction. IEEE Trans. Ind. Inform., vol. 18, no. 4, pp. 2830–2839, 2021.
[11] Yang B, Taleb T, Fan Y, et al. Mode selection and cooperative jamming for covert communication in d2d underlaid uav networks. IEEE Network 2021; **35**: 104–111.
[12] Jadav NK, Rathod T, Gupta R, et al. Blockchain-based secure and intelligent data dissemination framework for uavs in battlefield applications. IEEE Commun Stand Mag 2023; **7**: 16–23
[13] Saraswat D, Bhattacharya P, Singh A, et al. Secure 5g-assisted uav access scheme in iobt for region demarcation and surveillance operations. IEEE Commun Stand Mag 2022; **6**: 58–66
[14] Simmons GJ. The prisoners' problem and the subliminal channel. In: Proc. CRYPTO, 1984, 51–67.
[15] Luo Y, Qin J, Xiang X, et al. Coverless real-time image information hiding based on image block matching and dense convolutional network. J Real-Time Image Process 2020; **17**: 125–135.
[16] Peng F, Lin Z, Zhang X, et al. Reversible data hiding in encrypted 2d vector graphics based on reversible mapping model for real numbers. IEEE Trans Inf Forensics Secur 2019; **14**: 2400–2411.
[17] Long M, Peng F and Li H-y. Separable reversible data hiding and encryption for hevc video. J Real-Time Image Process 2018; **14**: 171–182.
[18] Liao X, Yu Y, Li B, et al. A new payload partition strategy in color image steganography. IEEE Trans Circuits Syst Video Technol 2019; **30**: 685–696.
[19] Wang Z, Feng, Shen L, et al. Cover selection for steganography using image similarity. IEEE Trans Dependable Secur Comput 2022; **20**: 920–935.
[20] Qiao T, Luo X, Wu T, et al. Adaptive steganalysis based on statistical model of quantized dct coefficients for jpeg images. IEEE Trans Dependable Secur Comput 2019; **18**: 2736–2751.
[21] Zhang Y, Luo X, Wang J, et al. Image robust adaptive steganography adapted to lossy channels in open social networks. Inf Sci 2021; **564**: 306–326.
[22] Mohsin AH, Zaidan A, Zaidan B, et al. Pso–blockchain-based image steganography: towards a new method to secure updating and sharing covid-19 data in decentralised hospitals intelligence architecture. Multimed Tools Appl 2021; **80**: 14 137–14 161.
[23] Ma K, Zhang W, Zhao X, et al. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inf Forensics Secur 2013; **8**: 553–562.
[24] Sharifzadeh M, Aloraini M and Schonfeld D. Adaptive batch size image merging steganography and quantized gaussian image steganography. IEEE Trans Inf Forensics Secur 2019; **15**: 867–879.
[25] Simmons GJ. Subliminal communication is easy using the dsa. In: Proc. EUROCRYPT, 1993, 218–232.
[26] Anderson R, Vaudenay S, Preneel B, et al. The newton channel. In: Proc. IH, 1996, 151–156.
[27] Bohli J-M, González Vasco MI and Steinwandt R. A subliminal-free variant of ecdsa. In: Proc. IH, 2007, 375–387.
[28] Jan J-K and Tseng Y-M. New digital signature with subliminal channels based on the discrete logarithm problem. In: Proc. IEEE CMC, 1999, 198–203.
[29] Bernstein DJ, Duif N, Lange T, et al. High-speed high-security signatures. J Cryptogr Eng 2012; **2**: 77–89.
[30] Hartl A, Annessi R and Zseby T. A subliminal channel in eddsa: Information leakage with high-speed signatures. In: Proc. ACM CCS, 2017, 67–78.
[31] Li Y, Ding L, Wu J, et al. Research on a new network covert channel model in blockchain environment. J Commun 2019; **40**: 67–79.
[32] Partala J. Provably secure covert communication on blockchain. Cryptography 2018; **2**: 18.
[33] Zhang P, Cheng Q, Zhang M, et al. A group covert communication method of digital currency based on blockchain technology. IEEE Trans Network Sci Eng 2022; **9**: 4266–4276.
[34] Zhang L, Zhang Z, Wang W, et al. Research on a covert communication model realized by using smart contracts in blockchain environment. IEEE Syst J 2021; **16**: 2822–2833.
[35] Zhang L, Zhang Z, Wang W, et al. A covert communication method using special bitcoin addresses generated by vanitygen. Comput Mat Contin 2020; **65**: 597–616.
[36] Torki O, Ashouri-Talouki M and Mahdavi M. Blockchain for steganography: Advantages, new algorithms and open challenges. In: Proc Int ISC Conf Inf Secur Cryptol, 2021, 1–5.
[37] Xu M, Wu H, Feng G, et al. Broadcasting steganography in the blockchain. In: Proc. IWDW, 2020, 256–267.
[38] Alsalami N and Zhang B. Uncontrolled randomness in blockchains: Covert bulletin board for illicit activity. In: Proc. IEEE IWQoS, 2020, 1–10.
[39] Cao H, Yin H, Gao F, et al. Chain-based covert data embedding schemes in blockchain. IEEE Internet Things J 2020; **9**: 14 699–14 707.
[40] Chen Z, Zhu L, Jiang P, et al. Blockchain meets covert communication: A survey. IEEE Commun Surv Tutorials 2022; **24**: 2163–2192.
[41] Gao F, Zhu L, Gai K, et al. Achieving a covert channel over an open blockchain network. IEEE Network 2020; **34**: 6–13.
[42] Young A and Yung M. The prevalence of kleptographic attacks on discrete-log based cryptosystems. In: Proc. CRYPTO, 1997, 264–276.
[43] Tian J, Gou G, Liu C, et al. Dlchain: A covert channel over blockchain based on dynamic labels. In: Proc. ICICS, 2020, 814–830.
[44] Zhang P, Cheng Q, Zhang M, et al. A blockchain-based secure covert communication method via shamir threshold and stc mapping. IEEE Trans Dependable Secur Comput 2024.
[45] Basuki AI and Rosiyadi D. Joint transaction-image steganography for high capacity covert communication. In: Proc. IC3INA, 2019, 41–46.
[46] Liu S, Fang Z, Gao F, et al. Whispers on ethereum: Blockchain-based covert data embedding schemes. In: Proc. ASIACCS, 2020, 171–179.

[47] Frkat D, Annessi R and Zseby T. Chainchannels: Private botnet communication over public blockchains. In: Proc. IEEE CPSCom, 2018, 1244–1252.
[48] Biryukov A, Feher D and Vitto G. Privacy aspects and subliminal channels in zcash. In: Proc. ACM CCS, 2019, 1813–1830.
[49] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system, 2008.
[50] Bonneau J, Miller A, Clark J, et al. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In: Proc. IEEE S & P, 2015, 104–121.
[51] Shen M, Tang X, Zhu L, et al. Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities. IEEE Internet Things J 2019; **6**: 7702–7712.
[52] Zhang Y, Xu C, Cheng N, et al. Chronos^{+}: An accurate blockchain-based time-stamping scheme for cloud storage. IEEE Trans Serv Comput 2019; **13**: 216–229.
[53] Zhang Y, Xu C, Lin X, et al. Blockchain-based public integrity verification for cloud storage against procrastinating auditors. IEEE Trans Cloud Comput 2019; **9**: 923–937.
[54] Li S, Zhang Y, Xu C, et al. Healthfort: A cloud-based ehealth system with conditional forward transparency and secure provenance via blockchain. IEEE Trans Mob Comput 2022; **22**: 6508–6525.
[55] Shamir A. How to share a secret. Commun ACM 1979; **22**: 612–613.
[56] Wang Y, Chen R, Huang X, et al. Sender-anamorphic encryption reformulated: Achieving robust and generic constructions. In: Proc. ASIACRYPT, 2023, 135–167.
[57] Von Ahn L and Hopper NJ. Public-key steganography. In: Proc. EUROCRYPT, 2004, 323–341.
[58] Kappos G, Yousaf H, Maller M, et al. An empirical analysis of anonymity in zcash. In: Proc. USENIX Security, 2018, 463–477.
[59] Hopper NJ, Langford J and Von Ahn L. Provably secure steganography. In: Proc. CRYPTO, 2002, 77–92.

Weijia Li received his B.Sc. degree from the University of Electronic Science Technology of China (UESTC), China, in 2022. He is currently a master student in the School of Computer Science and Engineering (School of Cyber Security) at the University of Electronic Science Technology of China. His research interests are applied cryptography, data security, and blockchain technology.

Yuan Zhang received his B.Sc. and Ph.D. degrees from the University of Electronic Science Technology of China (UESTC), China, in 2013 and 2019, respectively. He was a Visiting Ph.D. Student with BBCR Lab, Department of ECE, University of Waterloo, Canada, from 2017 to 2019. He is currently an Assistant Professor at the School of Computer Science and Engineering at UESTC. His research interests include applied cryptography, data security, and blockchain technology.

Xinyu He received her M.E. degree from Xidian University, China, in 2022. She is currently working toward a Ph.D. degree in the School of Computer Science and Engineering (School of Cybersecurity) at the University of Electronic Science Technology of China. Her research interests include applied cryptography, data exchange, and data security.

Yaqing Song received her B.Sc. degree from the University of Electronic Science Technology of China (UESTC), China, in 2021. She is currently a master student in the School of Computer Science and Engineering at the University of Electronic Science and Technology of China. Her research interests are applied cryptography and data security.