IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 1, 2021, accepted February 15, 2021, date of publication February 18, 2021, date of current version March 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3060285

MRCC: A Practical Covert Channel Over Monero With Provable Security

ZHAOZHONG GUO[1], LIUCHENG SHI[1], MAOZHI XU[1], AND HONG YIN[2]
[1]School of Mathematical Sciences, Peking University, Beijing 100871, China
[2]Institute of Computing Technology, China Academy of Railway Sciences, Beijing 100081, China

Corresponding author: Maozhi Xu (mzxu@math.pku.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62072011 and Grant 61672059.

ABSTRACT Covert channels are designed to protect the communication relationship of the sender and receiver. Traditional covert channels have become insecure due to the continuous improvement of traffic analysis techniques. In this context, there is an urgent need to identify new approaches for covert channels. Blockchain is an emerging technique with characteristics of user anonymity, a flooding propagation mechanism, and tamper resistance, which make it a compelling platform for covert channels. Previous approaches applied Bitcoin as the underlying blockchain, and its pseudoanonymity may expose the communication relationship. Moreover, the reliance of these approaches on prenegotiated labels to identify transactions containing covert messages further reduced their concealment. In this work, we present a practical and secure covert channel over Monero. Compared to Bitcoin, Monero's full anonymity efficiently protects the relationship between the sender and receiver. Moreover, no labels are employed to identify special transactions. The receiver filters and extracts the covert message using his private key. In this study, we make a complete assessment of the robustness, reliability, and anti-traceability of our protocol, as these properties are regarded as desirable for a covert channel. We also formalize the definition of security for covert channels through a transaction distinguishing experiment. A rigorous proof shows that our protocol meets this definition and is secure to use. Finally, we make a detailed comparison between our protocol and the existing blockchain-based covert channels.

INDEX TERMS Covert channel, blockchain, anonymity, label, provable security.

I. INTRODUCTION

Covert channels, which aim to protect the relationship between the sender and receiver by hiding the existence of secret communication, provide reliable privacy in sensitive scenarios, such as military and government communication, that suffer from attacks seeking to steal data from private organizations [1]–[3]. There are two primary types of covert channels [4], i.e., covert timing channels (CTCs) and covert storage channels (CSCs). CTCs, which hide covert messages in timing behavior, are significantly influenced by network delays or jitters; therefore, they have poor robustness. CSCs hide covert messages in storage fields, which can be detected through pattern categorization [5] and tampered with via normalization [6], [7]. A practical covert channel requires robustness, reliability, anti-traceability, and undetectability, which cannot be fully satisfied by traditional covert channels due to the continuous improvement of traffic analysis techniques [8]–[10]. Therefore, there is an urgent need to identify new approaches for covert channels.

Since Bitcoin [11] was designed and implemented in 2009, blockchain has become a platform with high credibility and reliability for providing data authenticity without any centralized parties. A blockchain's network is free for a participant to join, and a digital account is used instead of one's real-world identity to provide user anonymity. Moreover, the flooding propagation mechanism ensures that transactions can always be delivered from the sender to the receiver without direct communication. The characteristics of openness, anonymity, and tamper resistance make blockchain a compelling platform for constructing covert channels.

In recent years, a substantial amount of research seeking to achieve a covert channel over public blockchains has been published by academic researchers. According to previous approaches [14]–[17], the sender starts the data transmission by embedding the hidden message in a special transaction, which will be sent to the blockchain network and recorded in a block. Next, the receiver scans the blockchain and identifies

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

the special transaction from a flood of normal transactions using a prenegotiated label that usually appears in the form of a sending address or a receiving address. Finally, the receiver extracts the message from the special transaction.

Most of the existing blockchain-based covert channels are built on Bitcoin and use the same sending or receiving address in each data transmission. Current research shows that Bitcoin is pseudoanonymous [12] and that address reuse is generally considered to be a bad practice since it leads to identity exposure. Thus, the existing covert channels have a low level of user anonymity, which may reveal the communication relationship between the sender and the receiver. None of the previous approaches addresses this risk. Another problem lies in the use of labels, which ensures the receiver's successful identification of the special transactions but may also help the attacker identify the covert channels through characteristic analysis [13].

These two drawbacks reduce the practicality of the existing blockchain-based covert channels and motivate our work.

A. CONTRIBUTIONS

Our contributions are summarized as follows:

- We propose a blockchain-based covert channel with two main innovations. The first innovation is that our protocol achieves a high level of user anonymity by applying Monero as the underlying blockchain. Compared to Bitcoin, Monero's full anonymity improves our protocol's concealment by hiding the communication relationship between the message sender and receiver. Moreover, its complex transaction structure contributes to a high hiding capacity by offering more opportunities to embed covert messages. The second innovation is that our protocol is label-less. In our protocol, the receiving address of the special transaction is generated according to the receiver's public key. The generation algorithm ensures that it is computationally indistinguishable from random addresses without knowledge of the receiver's private key. Thus, the receiver can use his private key to filter the special transactions, rather than prenegotiated labels.
- We perform a complete assessment of the robustness, reliability, and anti-traceability, which are regarded as desirable properties of a covert channel, of our protocol. The results show that our protocol achieves these properties. Therefore, the sender and receiver can communicate via our covert channel with high reliability and privacy.
- We present a formal definition of security for covert channels through a transaction distinguishing experiment. Then, we provide a rigorous proof showing that our protocol meets the definition and is secure to use in scenarios in which information needs to be transmitted covertly.
- We make a detailed comparison between our protocol and the existing blockchain-based covert channels in terms of hiding capacity, user anonymity, label usage, and security, which demonstrates our protocol's practicality.

B. PAPER ORGANIZATION

The remainder of the paper is organized as follows. In Sec. II, we overview the existing blockchain-based covert channel protocols. Sec. III presents the preliminaries. A detailed description of our protocol is provided in Sec. IV. In Sec. V, we assess the protocol and give a strict proof showing its security. We compare our protocol to the existing covert channels in Sec. VI and make a conclusion in Sec. VII.

II. RELATED WORK

There are four main existing blockchain-based covert channels: Blockchain Covert Channel (BLOCCE) [14], Chain Channel [15], DLchain [16], and Kleptography-based Covert Channel (KBCC) [17]. Their differences lie in two aspects: how they embed covert messages into blockchain transactions and how they filter special transactions carrying covert messages from normal transactions.

Regarding the first aspect, BLOCCE applies a receiving address to convey a covert message. The sender generates several transactions of which the least significant bits of the receiving addresses form the covert message. The fact that only a single bit is transferred in one transaction makes BLOCCE inefficient. One way to improve the transfer capacity of BLOCCE is to match multiple bits of the receiving address. However, the computational effort increases exponentially as the bit number increases. Both Chain Channel and DLchain employ the subliminal channel technique to embed covert messages into transaction signatures [18], [19]. Chain Channel substitutes the nonce used in ECDSA [20] with the covert message, whereas DLchain substitutes the private key with the covert message. In KBCC, the covert message is encrypted and embedded in the default storage parameters of the transaction, such as the OP_RETURN parameter in Bitcoin and the Input_Data parameter in Ethereum. The decryption key is hidden in the transaction signature through the kleptography technique [22].

In terms of the second aspect, BLOCCE and Chain Channel employ the sender's address as the fixed label to enable the receiver to identify special transactions from normal transactions. KBCC uses the receiver's address as the fixed label and combines it with the private key of kleptography in transaction filtering. As a comparison, DLchain employs a dynamic label, which is generated based on the statistical distribution of the real transaction data, to ensure concealment. To enable the receiver to extract covert messages from special transactions, the sender is required to send two distinct transactions with signatures using the same nonce, which may raise suspicion and help an adversary detect the covert channel, especially given that blockchains are constantly monitored for nonce reuses to discover private keys from the duplicates and steal the associated coins [21].

III. PRELIMINARIES

A. SECURITY ASSUMPTIONS

1) DECISIONAL DIFFIE–HELLMAN ASSUMPTION

Let $\mathbb{G}$ be a cyclic group generated by G whose order is a prime q. For $a, b, r \in Z_q^*$, given $A = a \cdot G$ and $B = b \cdot G$, it is difficult for a Probabilistic Polynomial Time (PPT) adversary to distinguish between $(ab) \cdot G$ and $r \cdot G$.

B. CRYPTOGRAPHY TOOLS

1) PUBLIC-KEY ENCRYPTION SCHEME WITH PSEUDORANDOM CIPHERTEXTS

A public-key encryption scheme $PKE = \{Gen, Enc, Dec\}$ has pseudorandom ciphertexts under a chosen-plaintext attack if a PPT adversary is unable to distinguish the ciphertext from a uniformly random string, even if he was able to choose the plaintext [23]. To give a formal definition, we first describe a pseudorandom ciphertext experiment with a two-stage PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in Algorithm 1. In the first stage, $\mathcal{A}_1$ is given oracle access to Enc under a random public key PK_e and outputs a plaintext M. In the second stage, based on a coin toss, $\mathcal{A}_2$ is run with either the ciphertext of M or a random string. The success advantage of $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{PRC}_{\mathcal{A},PKE}(s) = \left| 1/2 - Pr[PRC_{EXP}{}^{PKE}_{\mathcal{A}}(1^s) = 1] \right|.$$

PKE has pseudorandom ciphertexts under a chosen plaintext attack if $\mathbf{Adv}^{PRC}_{\mathcal{A},PKE}(s)$ is negligible for every PPT adversary.

Algorithm 1 Ciphertext Distinguishing Experiment

procedure $PRC\_EXP^{PKE}_{\mathcal{A}}(1^s)$
  $(PK_e, SK_e) \leftarrow Gen(1^s)$
  $M \leftarrow \mathcal{A}_1^{Enc_{PK_e}}(1^s)$
  $r \leftarrow_R \{0, 1\}$
  if $r = 1$ **then**
    $C \leftarrow Enc(PK_e, M)$
  else
    $C \leftarrow_R (\{0, 1\}^{|Enc(PK_e, M)|})$
  end if
  $r' \leftarrow \mathcal{A}_2^{Enc}(C)$
  if $r = r'$ **then**
    return 1
  else
    return 0
  end if
end procedure

2) RING SIGNATURE

The ring signature was first introduced by Rivest *et al.* in [24] to provide anonymity for the signer. This signature allows the signer to sign a message on behalf of a group of users. The signature is checked by a set of public keys, rather than the signer's own single public key. The verifier is convinced that the real signer is a member of the group but cannot identify him or her from the other members. Specifically, the ring signature's signing and verification algorithm are described as follows:

- $RingSign(PK[n], sk, m) = \sigma$, where $PK[n]$ is a group of n public keys in which the signer is a member, sk is the signer's private key, m is the message to be signed, and σ is the valid ring signature.
- $RingVerify(PK[n], m, \sigma) = result$, where $result = True$ if σ is valid; otherwise, $result = False$.

3) STEALTH ADDRESS

A stealth address [25] is a technique widely used in blockchain systems to hide the real receiving address of a transaction. Specifically, the sender generates a one-time address (OTA) based on the receiver's real address as the receiving address of a transaction. The receiver can identify and control the one-time address using his or her private key. The technical details for generating and verifying a one-time address are shown in Algorithm 2.

Algorithm 2 OTA Generation and Verification

G is the base point of an elliptic curve group whose order is a prime q, and $Hash$ denotes a secure hash function. The receiver's real address is denoted as (A, B) with the private key (a, b), where $A = a \cdot G$ and $B = b \cdot G$.

procedure $GenOTA(A, B)$
  $s \longleftarrow_R Z_q^*$
  $S \leftarrow s \cdot G$
  $R \leftarrow A + (Hash(s \cdot B)) \cdot G$
  return (R, S)
end procedure
procedure $VerOTA(OTA, b)$
  $(R, S) \leftarrow OTA$
  $R' = A + (Hash(b \cdot S)) \cdot G$
  if $R' = R$ **then**
    return $True$
  else
    return $False$
  end if
end procedure

C. A SIMPLE MODEL OF MONERO

Monero [26], [27] is a digital currency designed to provide a new level of privacy compared to Bitcoin's pseudoanonymity by hiding the amounts, origins, and destinations of transactions. For the sake of simplicity, we apply a simple model of Monero that abstracts away the details that are not related to our protocol.

1) USER ADDRESS

In Monero, a user has a dual-public-key address with the structure of (A, B), where $A = a \cdot G$, $B = b \cdot G$, and G is the base point of an elliptic curve group. a is called the spending key, and b is called the tracking key. We refer to A as the user's public key and (A, B) as the user's address.

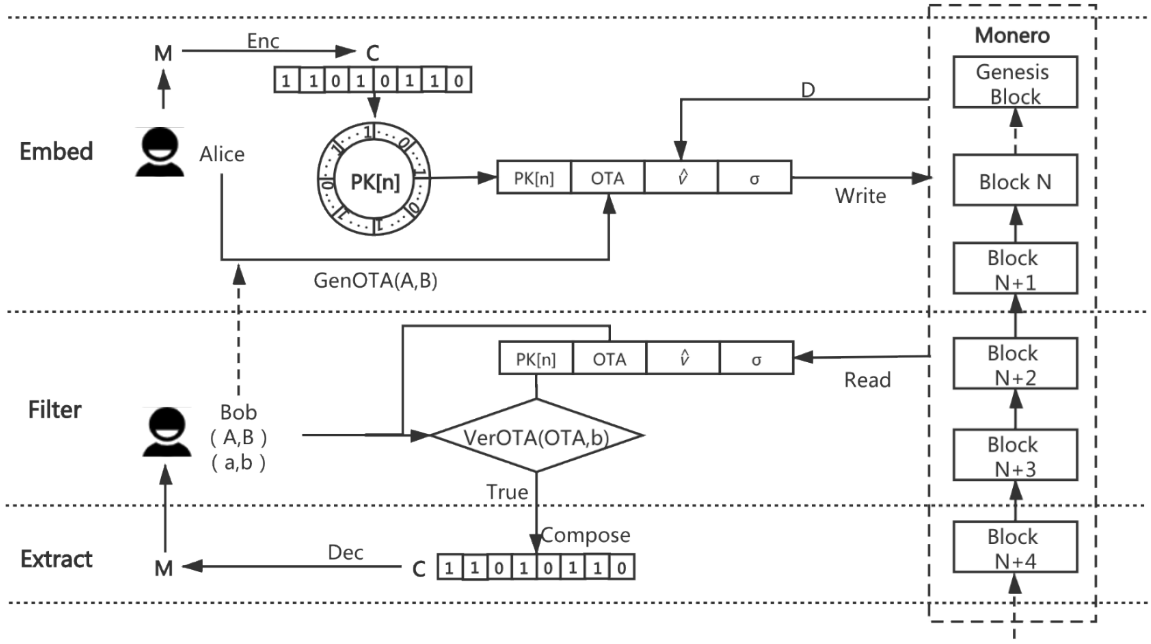

FIGURE 1. MRCC protocol.

2) TRANSACTION STRUCTURE

Monero uses the ring signature to provide anonymity for the transaction sender and stealth address to provide anonymity for the receiver. Moreover, the transaction's amount is hidden through a technique called confidential transactions [28]. Then, a transaction sent from (A, B) to (C, D) with amount v has the following form:

$$tx = (PK\,[n]\,, OTA, v, \sigma),$$

where $PK\,[n]$ is a set of public keys with A as a member in the ring signature, $OTA = GenOTA(C, D)$, and $\sigma = RingSign(PK\,[n]\,, a, (PK\,[n]\,, OTA, v))$.

3) ACCESS TO MONERO'S BLOCKCHAIN

We denote Monero's blockchain as $\mathcal{MB}$, which is publicly accessible. Anyone can write data to $\mathcal{MB}$ and read data from $\mathcal{MB}$. The two processes are abstracted as follows:

- $Write\,(tx, \mathcal{MB})$, where tx is a valid transaction that will be stored in $\mathcal{MB}$ permanently.
- $Read(\mathcal{MB})$, through which anyone can read all the transactions stored in $\mathcal{MB}$.

IV. PROTOCOL DESCRIPTION

In this section, we provide a detailed description of our protocol called MRCC (Monero-based Covert Channel). We use Monero as the underlying blockchain of our protocol. The sender starts the data transmission by sending a special transaction, of which the public key set is used for message embedding and the receiving address is used for transaction filtering. Specifically, the sender selects a set of public keys and orders them such that their least significant bits (LSBs) form the covert message. The receiver identifies the special transaction by checking the receiving address with his private key through the OTA verification algorithm and then extracts the covert message.

In the remainder of this section, we first give a general overview of the protocol and show its workflow. Next, we present the technical details of message embedding, transaction filtering, and message extraction separately.

A. GENERAL OVERVIEW

Imagine a scenario in which Alice attempts to convey a message M covertly to Bob through the Monero blockchain. Before transmission, both parties negotiate some key information that is necessary for transmission, including a public-key encryption scheme $PKE = (Gen, Enc, Dec)$, a related key pair (PK_e, SK_e) for message encryption and decryption, and Bob's address (A, B). We note that the encryption scheme PKE has pseudorandom ciphertexts under a chosen-plaintext attack. For the sake of simplicity, we assume the bit length of M and its ciphertext under PKE is n. The protocol proceeds as follows:

- Step 1: Alice encrypts M using PK_e and obtains C as the ciphertext.
- Step 2: Alice constructs a special transaction $tx = (PK\,[n]\,, OTA, v, \sigma)$, where $PK[n]$ is a public key set with C embedded in it, OTA is Bob's one-time address, and σ is a valid ring signature generated by Alice's spending key.
- Step 3: Alice writes tx in Monero's blockchain.
- Step 4: Bob scans Monero's blockchain and identifies tx from the normal transactions using his tracking key.
- Step 5: Bob extracts $PK\,[n]$ from tx and forms C by taking the least significant bit of each public key's hash value. Then, he decrypts C using SK_e and obtains M as output.

An overview of the protocol is depicted in Fig. 1.

B. MESSAGE EMBEDDING

Alice conveys M covertly to Bob by constructing a special transaction $tx = (PK[n], OTA, v, \sigma)$, which has the same format as normal transactions. The components of tx need to be constructed properly to ensure the concealment of tx so that the adversary cannot distinguish it from normal transactions.

– Construction of $PK[n]$

We denote Alice's address set as Acc, which is divided into two subsets, i.e., Acc_0 and Acc_1, as follows:

$$Acc_0 = \{(SPK_i^0, ssk_i^0) | LSB(Hash(SPK_i^0)) = 0, i \leq m_0\}$$
$$Acc_1 = \{(SPK_i^1, ssk_i^1) | LSB(Hash(SPK_i^1)) = 1, i \leq m_1\}.$$

The set of public keys drawn from the transaction history of $\mathcal{MB}$ is denoted as Pub, which is also divided into two subsets, i.e., Pub_0 and Pub_1:

$$Pub_0 = \{RPK_i^0 | LSB(Hash(RPK_i^0)) = 0, i \leq t_0\}$$
$$Pub_1 = \{RPK_i^1 | LSB(Hash(RPK_i^1)) = 1, i \leq t_1\}.$$

Alice starts the construction by encrypting M into ciphertext C with bit representation $c_0c_1 \cdots c_{n-1} \in \{0,1\}^n$. From Sec. III-C, we know that only one public key from $PK[n]$ belongs to Alice. Without any loss of generality, we assume that $PK[p]$ is Alice's public key. $PK[p]$ will be randomly selected from Acc_0 if $c_p = 0$; otherwise, it will be randomly selected from Acc_1. For the other $n-1$ positions, $PK[i]$ is randomly selected from Pub_0 if $c_i = 0$; otherwise, it is randomly selected from Pub_1.

Clearly, the least significant bits of the public keys' hash values form C:

$$LSB(Hash(PK[i])) = c_i, \quad \text{for } 0 \leq i < n.$$

We note that $PK[n]$ is randomly selected according to C and thus different in each data transmission.

– Construction of OTA

OTA is a one-time address for Bob generated by Alice through $GenOTA$.

– Construction of v

v is selected according to $\mathcal{D}(PK[p])$, which denotes the probability distribution of the value of $PK[p]$'s historic transactions. We note that it is important that v be selected according to this distribution to prevent the adversary from detecting the communication.

– Construction of σ

σ is the valid ring signature for tx generated by Alice's spending key.

Finally, tx is broadcast, mixed into normal transactions, and permanently stored in $\mathcal{MB}$.

The technical details of the message embedding are shown in Algorithm 3. We remark that our protocol's throughput can be improved to hundreds of bits per transaction by increasing the number of inputs in the special transaction.

Algorithm 3 Embedding Algorithm

```
procedure Embed(M, PK_e, (A, B))
    C ← Enc(PK_e, M)
    c_0c_1 ··· c_{n-1} ← Interpret(C)
    p ←_R Z_n
    if c_p = 0 then
      (SPK, ssk) ←_R Acc_0
    else (SPK, ssk) ←_R Acc_1
    end if
    PK[p] ← SPK
    for 0 ≤ i < n do                    ▷ Construct PK[n]
        if i = p then
            continue
        else
            if c_i = 0 then
                PK[i] ←_R Pub_0
            else PK[i] ←_R Pub_1
            end if
        end if
    end for
    OTA ← GenOTA(A, B)                  ▷ Construct OTA
    v ← D(SPK)                          ▷ Construct v
    data_tx ← (PK[n], OTA, v)
    σ ← RingSign(PK[n], ssk, data_tx)   ▷ Sign tx
    tx ← (PK[n], OTA, v, σ)
    Write(tx, MB)
end procedure
```

C. TRANSACTION FILTERING

Bob reads transactions from $\mathcal{MB}$ and identifies the special transactions from normal transactions based on the OTA field. For each transaction tx_i read from $\mathcal{MB}$, Bob extracts OTA_i and verifies it using his tracking key b. tx_i is a special transaction if and only if OTA_i passes the verification. We note that even an adversary with sufficient time and computing power cannot filter a special transaction from normal transactions without the knowledge of b, which is secretly kept by Bob.

The technical details of transaction filtering are shown in Algorithm 4. Clearly, no fixed labels are used in this procedure.

Algorithm 4 Transaction Filtering Algorithm

```
procedure Filter(MB, b)
    TxSet ← ∅
    tx_1, tx_2, ..., tx_s ← Read(MB)
    for 1 ≤ i ≤ s do
        (PK_i[n], OTA_i, v_i, σ_i) ← tx_i
        if VerOTA(OTA_i, b) = True then
            TxSet ← TxSet ∪ {tx_i
        else continue
        end if
    end for
    return TxSet
end procedure
```

D. MESSAGE EXTRACTION

Message extraction is straightforward. Bob extracts $PK[n]$ from the special transaction and forms C by composing the

least significant bit of each public key's hash value. Finally, he decrypts C using SK_e and obtains M as the output.

The technical details of the message extraction are shown in Algorithm 5.

Algorithm 5 Extraction Algorithm

procedure $Extract(tx, SK_e)$
  $(PK[n], OTA, v, \sigma) \leftarrow tx$
  for $0 \leq i < n$ **do**
    $c_i = LSB(Hash(PK[i]))$
  end for
  $C \leftarrow Compose(c_0 c_1 \cdots c_{n-1})$
  $M \leftarrow Dec(SK_e, C)$
  return M
end procedure

The variables and functions used in *Embed*, *Filter*, and *Extract* are collected in Table 1 for easy reference.

TABLE 1. Used variables and functions.

Notation	Definition
M	Message to be sent by Alice to Bob
$\mathcal{MB}$	Monero's blockchain
$Write()$	Write transactions to $\mathcal{MB}$
$Read()$	Read transactions from $\mathcal{MB}$
(A, B)	Bob's address
b	Bob's tracking key
PKE	A public-key encryption scheme with pseudorandom ciphertexts
$Hash()$	Secure hash function
$Enc()$	Encryption algorithm of PKE
$Dec()$	Decryption algorithm of PKE
(PK_e, SK_e)	A prenegotiated keypair of PKE
C	The ciphertext of M: $C = Enc(PK_e, M)$
$LSB()$	The least significant bit of the input
Acc	Alice's address set:$Acc = Acc_0 \cup Acc_1$
Acc_0	The set of Alice's addresses with the least significant bits of their hash values equal to 0
Acc_1	The set of Alice's addresses with the least significant bits of their hash values equal to 1
Pub	The set of public keys drawn from the transaction history of $\mathcal{B}$: $Pub = Pub_0 \cup Pub_1$
Pub_0	The subset of public keys from Pub with the least significant bits of their hash values equal to 0
Pub_1	The subset of public keys from Pub with the least significant bits of their hash values equal to 1
$\mathcal{D}()$	The probability distribution of the transaction value of a specific public key
$Interpret()$	Convert text format to bit representation format
$Compose()$	Convert bit representation format to text format
$GenOTA()$	Stealth address generation algorithm
$VerOTA()$	Stealth address verification algorithm
$RingSign()$	Signing algorithm of Monero's ring signature scheme

V. ASSESSMENT AND SECURITY ANALYSIS

In this section, we perform a complete assessment of the robustness, reliability, and anti-traceability of our algorithm. Moreover, we follow the approach of [14] by building a security model for blockchain-based covert channels and present a rigorous proof showing that our protocol is secure by achieving high concealment.

We note that the secure hash function used in our protocol is modeled as a random oracle whose output is uniformly distributed. Besides, digital signatures are existentially unforgeable.

A. ROBUSTNESS

Robustness means that the receiver can always receive special transactions and extract the covert messages correctly, even in the presence of an adversary. Our protocol's robustness is illustrated in two aspects, i.e., its tamper resistance and the soundness of its extraction. The former ensures the integrity of the special transaction whereas the latter ensures the correctness of the extracted data.

1) TAMPER RESISTANCE

Due to the default flood propagation mechanism applied by the blockchain network, a block containing the special transaction will eventually be included in the blockchain. Therefore, tampering with the special transaction equals tampering with the related block. According to previous approaches [29], the probability of successfully tampering with a block decreases exponentially as its depth in the blockchain increases. Therefore, the special transaction is tamper-resistant, which means that the receiver can always receive the correct transaction data sent from the sender.

2) SOUNDNESS OF EXTRACTION

The soundness of its extraction is guaranteed by the soundness of the encryption scheme PKE. As long as the ciphertext embedded in the special transaction is not tampered with, the receiver can always decrypt it using the private key SK_e and obtain the correct plaintext.

B. RELIABILITY

Reliability means that the probability of a normal transaction being filtered as a special transaction by the receiver is negligible. To illustrate our protocol's reliability, we provide the following theorem:

Theorem 1: In the MRCC, a normal transaction is filtered as the special transaction by the receiver with probability $P < 4/q$, where q is the order of the elliptic curve group.

Proof: tx is a normal transaction with (R, S) as its one-time address. Assuming tx is intended to send to address (C, D), then from Algorithm 2, we have

$$\begin{cases} S = s \cdot G \\ R = C + (Hash(s \cdot D)) \cdot G \end{cases}. \tag{1}$$

Bob is the receiver of our protocol with address (A, B). According to Algorithm 4, Bob filters tx as the special

transaction if the following equations hold:

$$\begin{cases} S = s \cdot G \\ R = A + (Hash(s \cdot B)) \cdot G \end{cases}. \tag{2}$$

From Eq. 1 and Eq. 2, we have

$$\begin{aligned} P &= Pr[C + (Hash\,(s \cdot D)) \cdot G = A + (Hash\,(s \cdot B)) \cdot G] \\ &= Pr[a + (Hash\,(s \cdot B)) = c + (Hash\,(s \cdot D))\ mod\ q] \end{aligned}$$

Because the output of *Hash* is uniformly distributed, we have

$$P = Pr[t_1 = t_2\ mod\ q], \tag{3}$$

where both t_1 and t_2 are randomly selected from *Hash*'s output space Z_p. We remark that $p > q$ in Monero.

For a fixed value $x \in Z_q$, we have

$$\begin{aligned} \hat{P} &= Pr\,[t_1 = x\ mod\ q] \\ &\leq (\lfloor p/q \rfloor + 1)/p \leq 1/q + 1/p \end{aligned} \tag{4}$$

Then, P is computed as

$$\begin{aligned} P &\leq \hat{P}^2 \times q \\ &= 2/p + 1/q + q/p^2 \leq 4/q \end{aligned} \tag{5}$$

Since q is a 252-bit large number, we claim that P is negligible. ■

According to Theorem 1, MRCC achieves reliability.

C. ANTI-TRACEABILITY

Anti-traceability means that the adversary cannot associate the special transactions containing covert messages or perform statistical analysis to determine the relevance of the covert communication channel users. Our protocol achieves anti-traceability through the anonymity of the sender and receiver. The former ensures that for each incoming transaction, all possible senders are equiprobable; the latter ensures that it is impossible to prove that two outgoing transactions were sent to the same receiver.

1) SENDER'S ANONYMITY

Our protocol is designed based on Monero, which provides probabilistic anonymity for transaction senders through ring signatures. In Monero, the sending address of a transaction is a group of public keys. The adversary is convinced that the sender is a member of that group but cannot exclusively identify the sender. Moreover, the sender uses different public keys in each covert communication of our protocol, which further increases the difficulty of estimating the sender's identity.

2) RECEIVER'S ANONYMITY

In our protocol, a one-time address is generated from the receiver's original address in each covert transmission. This one-time address is set as the receiving address of the special transaction to protect the receiver's identity. The one-time address will not leak any information about the receiver. We prove this by showing that the one-time address is indistinguishable from a pair of random points on the elliptic curve.

Theorem 2: Given the receiver's address, it is difficult for a PPT adversary to distinguish between the receiver's one-time address and a pair of random points on the elliptic curve.

Proof: We show that if there is an adversary $\mathcal{A}'$ that can distinguish between the receiver's one-time address and a pair of random points on the elliptic curve, then there is an adversary $\mathcal{A}$ that can use $\mathcal{A}'$ to break the DDH assumption with the same advantage.

Let (G, A, B, T) be an instance of the DDH problem, where $A = a \cdot G$ and $B = b \cdot G$. The task of $\mathcal{A}$ is to guess whether $T = (ab) \cdot G$ or $T = r \cdot G$, where r is a random number. $\mathcal{A}$ performs as follows:

- $\mathcal{A}$ sends (A, B) to $\mathcal{A}'$ as the receiver's address.
- $\mathcal{A}$ computes $S = A$ and $R = A + (Hash(T)) \cdot G$ and sends (R, S) to $\mathcal{A}'$ as a one-time address for the receiver.
- If $\mathcal{A}'$ accepts (R, S) as a one-time address, then $\mathcal{A}$ guesses $T = (ab) \cdot G$; otherwise, $\mathcal{A}$ guesses $T = r \cdot G$

According to Algorithm 2, (R, S) constitutes a one-time address for (A, B) when $T = (ab) \cdot G$. Thus, $\mathcal{A}$ breaks the DDH assumption with the same advantage as $\mathcal{A}'$. ■

We remark that one-time addresses belonging to the same receiver have no common characteristic; thus, their relationship cannot be discovered. Therefore, the adversary cannot trace the covert communication by tracking the receiver.

D. SECURITY

We start by describing the threat model. In our scenario, Alice is the message sender with address set *Acc* who attempts to send a covert message M to the receiver Bob with address (A, B) through the Monero blockchain $\mathcal{MB}$. They agreed beforehand on a key pair (PK_e, SK_e) of the public-key encryption scheme *PKE* for message encryption and decryption. The PPT adversary attempts to detect the presence of covert communication on $\mathcal{MB}$ with the following settings:

- The adversary has the knowledge of Alice's address set *Acc*, Bob's address (A, B), and the encryption key PK_e. However, we do not give him access to Alice's and Bob's private keys or to the decryption key SK_e.
- The adversary determines the transmitted message M, which means it is a chosen hidden-text attack.
- The adversary has complete access to the blockchain $\mathcal{MB}$ but cannot block Alice from writing valid transactions to $\mathcal{MB}$ or prevent Bob from reading transactions in $\mathcal{MB}$.

We say that a covert channel over a public blockchain is secure if it achieves high concealment, which means that the special transactions carrying covert messages are indistinguishable from the normal transactions. We remark that security is also referred to as undetectability. To give a formal definition, we follow the approach of [14] by describing a transaction distinguishing experiment in which the task of the adversary is to determine whether the given transaction is a normal transaction or a special transaction with a

covert message embedded in it. Specifically, the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is modeled in two stages, i.e., $\mathcal{A}_1$ and $\mathcal{A}_2$. In the first stage, $\mathcal{A}_1$ is given full access to *Read*() and *Submit*() and outputs a message M. Based on a coin toss, either a special transaction carrying M according to our protocol or a normal transaction is generated and sent to the blockchain. In the second stage, $\mathcal{A}_2$ is invoked and outputs one bit seeking to distinguish whether there is a special transaction in the blockchain.

The details of the experiment are shown in Algorithm 6. For simplicity, $Gen_{\mathcal{MB}}(1^s)$ denotes the generation of Monero's address, and $GenNormTrans(1^s)$ denotes the generation of a random normal transaction.

Algorithm 6 Transaction Distinguishing Experiment

1: **procedure** $TXD\_EXP_{\mathcal{A}}^{CCP,\mathcal{MB}}(1^s)$
2: $(PK_e, SK_e) \leftarrow Gen(1^s)$
3: $Acc, (A, B) \leftarrow Gen_{\mathcal{MB}}(1^s)$
4: $M \leftarrow \mathcal{A}_1^{Read,Submit}(PK_e, (A, B), Acc)$
5: $r \leftarrow_R \{0, 1\}$
6: **if** $r = 1$ **then**
7: $Embed(M, PK_e, (A, B))$
8: **else**
9: $tx \leftarrow GenNormTrans(1^s)$
10: $Write(tx, \mathcal{MB})$
11: **end if**
12: $r' \leftarrow \mathcal{A}_2^{Read,Submit}(PK_e, (A, B), Acc)$
13: **if** $r = r'$ **then**
14: **return** 1
15: **else**
16: **return** 0
17: **end if**
18: **end procedure**

The advantage of an adversary in distinguishing the special transaction in the above experiment is defined as

$$\mathbf{Adv}_{\mathcal{A},CCP,\mathcal{MB}}^{TX-DIS}(s) = \left|1/2 - Pr[TXD\_EXP_{\mathcal{A}}^{CCP,\mathcal{MB}}(1^s) = 1]\right|$$

If this advantage is significantly greater than 0, then in practice, the adversary can detect the covert channel. For a secure protocol, it is desired that the advantage be negligible. In this context, we formalize the definition of a blockchain-based covert channel's security as below.

Definition 1: A blockchain-based covert channel is secure if the advantage in the transaction distinguishing experiment is negligible for every PPT adversary.

We show our protocol's security through the following theorem.

Theorem 2: MRCC is secure against a PPT adversary.

Proof: Suppose there is an adversary $\mathcal{A}' = (\mathcal{A}_1', \mathcal{A}_2')$ who succeeds in the transaction distinguishing experiment with non-negligible advantage. Then, we can construct an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ to achieve the same advantage in the ciphertext distinguishing experiment for the public-key encryption scheme *PKE*. Since *PKE* has pseudorandom ciphertexts under a chosen-plaintext attack and thus the advantage in the ciphertext distinguishing experiment is negligible for any PPT adversary, we come to our conclusion.

The adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is constructed according to Algorithm 7.

Algorithm 7 Ciphertext Distinguishing Experiment by $\mathcal{A}$

procedure $\mathcal{A}_1^{Enc}(1^s)$
$(PK_e, SK_e) \leftarrow Gen(1^s)$
$Acc, (A, B) \leftarrow Gen_{\mathcal{MB}}(1^s)$
$M \leftarrow \mathcal{A}_1'^{Read,Submit}(PK_e, (A, B), Acc)$
return M
end procedure
$\mathcal{A}_1$ chooses M as the plaintext in the ciphertext distinguishing experiment and is given C as the input for stage 2.
procedure $\mathcal{A}_2^{Enc}(C, PK_e)$
Embed C into $\mathcal{MB}$ by simulating *Embed*
$r' \leftarrow \mathcal{A}_2'^{Read,Submit}(PK_e, (A, B), Acc)$
return r'
end procedure

Clearly, $\mathcal{A}$ is a PPT adversary. Let R be the result of a coin toss (Algorithm 1, line 4) in the ciphertext distinguishing experiment. In the case of $R = 1$, $\mathcal{A}$ is given the correct ciphertext of M, i.e., $C = Enc(PK_e, M)$. According to our construction, $\mathcal{A}'$ is given the special transaction in the transaction distinguishing experiment in this case. Because the output of $\mathcal{A}$ is the same as $\mathcal{A}'$ (Algorithm 7, line 9), we have the following equation:

$$Pr[\mathcal{A}\ succeeds|R = 1] = Pr[\mathcal{A}'\ succeeds|R = 1].$$

Similarly, we have the following equation for the case of $R = 0$:

$$Pr[\mathcal{A}\ succeeds|R = 0] = Pr[\mathcal{A}'\ succeeds|R = 0].$$

From the two equations, we have

$$\mathbf{Adv}_{\mathcal{A},PKE}^{PRC}(s) = \mathbf{Adv}_{\mathcal{A}',CCP,\mathcal{MB}}^{TX-DIS}(s). \quad (6)$$

Because *PKE* has pseudorandom ciphertexts under a chosen-plaintext attack, both advantages in Eq. 6 are negligible. According to Definition 1, the MRCC is secure against a PPT adversary. ■

VI. COMPARISON

The covert channels share the common goal of transferring messages covertly between users with efficiency, anonymity, and security. In this section, we compare MRCC with the existing blockchain-based covert channels from the following aspects (a summary is provided in Table 2).

The hiding capacity is defined as the number of message bits transferred per transaction (bpt). A high hiding capacity means high efficiency. BLOCCE's hiding capacity is 1 bpt because it uses the least significant bit of a transaction's receiving address for message embedding. Chain Channel

TABLE 2. Comparison between MRCC and previous approaches.

Scheme	Hiding Capacity	User Anonymity	Label Usage	Provable Security
BLOCCE	1 bpt	Low	Yes	Yes
DLchain	256 bpt	Medium	Yes	No
Chain Channels	256 bpt	Low	Yes	No
KBCC	320 bpt	Low	Yes	No
MRCC	$11r$ bpt	High	No	Yes

and DLchain achieve a hiding capacity of 256 bpt by substituting the nonce or private key used in ECDSA signature generation with the covert message. KBCC inserts the hidden message in the OP_RETURN field that is up to 40 bytes in length, which results in a hiding capacity of 320 bpt. Monero applies the UTXO model [30], and multiple inputs can be included in a single transaction. Currently, the size of the ring signature's public key set in each input is 11. Therefore, MRCC's hiding capacity is calculated as $11r$, where r is the number of inputs in the special transaction. We can improve MRCC's hiding capacity to hundreds or even thousands bpt by simply increasing the number of inputs. MRCC's scalable hiding capacity contributes directly to its practicability.

User anonymity helps to hide the communication relationship between the sender and receiver by preventing transaction correlation attacks. An efficient way to achieve user anonymity is using different addresses in each covert communication. In this instance, we use it to measure the anonymity of covert channels. Chain Channel, BLOCCE, and KBCC have a low level of user anonymity because they all use the same sending or receiving address in each data transmission. As a comparison, DLchain and MRCC use different sending addresses and receiving addresses in each transmission. We remark that in MRCC, the sending address is further protected by a ring signature, which allows MRCC to achieve a higher level of user anonymity than DLchain.

Labels are commonly used in covert channels to ensure that the receiver identifies special transactions containing covert messages from thousands of normal transactions. Labels also increase the risk of channel exposure at the same time. In this context, a well-designed covert channel is expected to have less dependence on label usage. Both BLOCCE and Chain Channel use a fixed sending address as the label to filter the special transactions. DLchain uses dynamic labels whereas KBCC combines a fixed label with the private key of kleptography in transaction filtering. They all rely deeply on the use of labels. As a comparison, MRCC is a label-less covert channel in which the receiver identifies the special transaction by simply using his private key. We remark that this characteristic enables MRCC to achieve high concealment.

Security is a major consideration for covert channels in practical use. DLchain, Chain Channel, and KBCC assess their schemes from some specific aspects but have not presented formal security proofs. Following BLOCCE, we design a transaction distinguishing experiment and give a rigorous proof showing that MRCC is provably secure.

In summary, MRCC is a practical covert channel due to its scalable hiding capacity, high user anonymity, label-less transaction filtering process, and provable security.

VII. CONCLUSION

In this paper, we focus on the design of a covert channel over a public blockchain, which is urgently needed, since traditional covert channels have become insecure due to the continuous improvement of traffic analysis techniques. After intensive research into recently proposed protocols, we present MRCC, a practical covert channel over Monero with provable security. Compared with previous approaches, our protocol achieves high concealment and practicability. We also make a complete assessment of our protocol and provide a strict proof showing its security. We have not implemented MRCC in practice, which we leave as future work.

REFERENCES

[1] R. Deibert, R. Rohozinski, A. Manchanda, and N. Villeneuve. (2009). *Tracking GhostNet: Investigating a Cyber Espionage Network*. [Online]. Available: http://www.nartv.org/mirror/ghostnet.pdf
[2] S. Adair, R. Deibert, and R. Rohozinski. (2010). *Shadows in the Cloud: Investigating Cyber Espionage 2.0*. [Online]. Available: http://shadows-in-the-cloud.net
[3] E. Nakashima. (2014). *Identify Chinese Cyber Espionage Group*. [Online]. Available: https://tinyurl.com/pntdm64
[4] J. Millen, "20 years of covert channel modeling and analysis," in *Proc. IEEE Symp. Secur. Privacy*, Oakland, CA, USA, May 1999, pp. 113–114.
[5] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in Internet traffic with active wardens," in *Proc. 5th Int. Workshop Inf. Hiding (IH)*, Noordwijkerhout, The Netherlands, Dec. 2002, pp. 18–35.
[6] H. Mark, V. Paxson, and C. Kreibich, "Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics ur informatik unchen," *J. Soc. Mater. Sci. Jpn.*, vol. 46, no. 8, pp. 989–995, 2001, doi: 10.2472/jsms.46.989.
[7] G. Lewandowski, N. B. Lucena, and S. J. Chapin, "Analyzing network-aware active wardens in IPv6," in *Proc. 8th Int. Conf. Inf. Hiding (IH)*, Alexandria, VA, USA, Jul. 2006, pp. 58–77.
[8] M. Nasr, A. Bahramali, and A. Houmansadr, "DeepCorr: Strong flow correlation attacks on tor using deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 1962–1976.
[9] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless monster: Exploring the ecosystem of Web-based device fingerprinting," in *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, May 2013, pp. 541–555.
[10] D. Arp, F. Yamaguchi, and K. Rieck, "Torben: A practical side-channel attack for deanonymizing tor communication," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, Singapore, Apr. 2015, pp. 597–602.
[11] S. Nakamoto. (2008). *A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[12] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in zcash," in *Proc. 27th USENIX Conf. Secur. Symp. (SEC)*, Baltimore, MD, USA, Aug. 2018, pp. 463–477.
[13] R. Matzutt, M. Henze, J. H. Ziegeldorf, and J. Hiller, "Thwarting unwanted blockchain content insertion," in *Proc. 1st IEEE Workshop Blockchain Technol. Appl. (BTA)*, Orlando, FL, USA, Apr. 2018, pp. 364–370.
[14] J. Partala, "Provably secure covert communication on blockchain," *Cryptography*, vol. 2, no. 3, p. 18, Aug. 2018, doi: 10.3390/cryptography2030018.
[15] D. Frkat, R. Annessi, and T. Zseby, "ChainChannels: Private botnet communication over public blockchains," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Halifax, NS, Canada, Jul. 2018, pp. 1244–1252.

[16] J. Tian, G. Gou, C. Liu, Y. Chen, G. Xiong, and Z. Li, "DLchain: A covert channel over blockchain based on dynamic labels," in *Proc. 21st Int. Conf. Inf. Commun. Secur. (ICICS)*, Beijing, China, Dec. 2019, pp. 814–830.
[17] F. Gao, L. Zhu, K. Gai, C. Zhang, and S. Liu, "Achieving a covert channel over an open blockchain network," *IEEE Netw.*, vol. 34, no. 2, pp. 6–13, Mar. 2020, doi: 10.1109/MNET.001.1900225.
[18] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology*. Boston, MA, USA: Springer, 1984, pp. 51–67, doi: 10.1007/978-1-4684-4730-9_5.
[19] G. J. Simmons, "The subliminal channel and digital signatures," in *Proc. Workshop Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Paris, France, Apr. 1984, pp. 364–378.
[20] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001, doi: 10.1007/s102070100002.
[21] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *Proc. 18th Annu. Int. Conf. Financial Cryptograph. Data Secur. (FC)*, Christ Church, Barbados, Mar. 2014, pp. 157–175.
[22] A. Young and M. Yung, "Kleptography: Using cryptography against cryptography," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Konstanz, Germany, May 1997, pp. 62–74.
[23] B. Möller, "A public-key encryption scheme with pseudo-random ciphertexts," in *Proc. 9th Eur. Symp. Res. Comput. Secur. (ESORICS)*, Sophia Antipolis, France, Sep. 2004, pp. 335–351.
[24] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proc. 7th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*, Gold Coast, QLD, Australia, Dec. 2001, pp. 552–565.
[25] ByteCoin. *Untraceable Transactions Which Can Contain a Secure Message are Inevitable*. Accessed: Oct. 15, 2020. [Online]. Available: https://bitcointalk.org/index.php?topic=5965.0
[26] Monero. *About Monero*. Accessed: Aug. 12, 2020. [Online]. Available: https://www.getmonero.org/resources/about/
[27] S. Noether and A. Mackenzie, "Ring confidential transactions," Cryptol. ePrint Arch., Cambridge, U.K., Tech. Rep. 2015/1098. [Online]. Available: https://eprint.iacr.org/2015/1098
[28] G. Maxwell. *Confidential Transactions*. Accessed: Jun. 25, 2020. [Online]. Available: https://en.bitcoin.it/wiki/Confidential_transactions
[29] Y. Li, L. Ding, J. Wu, Q. Cui, X. Liu, and B. Guan, "Research on a new network covert channel model in blockchain environment," *J. Commun.*, vol. 40, no. 5, pp. 68–78, May 2019. Accessed: Oct. 12, 2020. [Online]. Available: http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2019111, doi: 10.11959/j.issn.1000-436x.2019111.
[30] SEBA Bank. (2020). *A Beginner's Guide to Blockchain Accounting Standards*. [Online]. Available: https://www.seba.swiss/research/A-Beginners-Guide-to-Blockchain-Accounting-Standards

ZHAOZHONG GUO received the B.S. degree in information and computing sciences from Peking University, China, in 2012, where he is currently pursuing the Ph.D. degree in applied mathematics. His current research interests include blockchain technology and public key cryptography and multiparty computation.

LIUCHENG SHI received the B.S. degree in information and computing sciences from Peking University, China, in 2012, where he is currently pursuing the Ph.D. degree in applied mathematics. His current research interests include blockchain technology and public key cryptography and applied cryptography.

MAOZHI XU received the B.S. degree in mathematics from Huaibei Normal University, China, in 1983, the M.S degree in mathematics from Wuhan University, China, in 1987, and the Ph.D. degree in mathematics from Peking University, China, in 1994. He is currently a Professor with the Peking University.

HONG YIN received the M.S. degree in computer science and technology from the National University of Defense Technology, China, in 2005. She is currently a Senior Engineer with the China Academy of Railway Sciences. Her current research interests include network security and applied cryptography.