IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Date of current version xxxx, 2018.

Digital Object Identifier 10.xxx/ACCESS.2018.DOI

A Blockchain based Truthful Incentive Mechanism for Distributed P2P Applications

YUNHUA HE[1], (Member,IEEE), HONG LI[2], (Member,IEEE), XIUZHEN CHENG[3], (Fellow,IEEE), YAN LIU[4],(Member,IEEE), CHAO YANG[5],(Member,IEEE), LIMIN SUN[2], (MEMBER,IEEE)

[1]Department of Computer Science, North China University of Technology, Beijing 100144 China
[2]Institute of Information Engineering, Chinese Academy of Science, Beijing 100093 China
[3]Department of Computer Science, The George Washington University, Washington DC 20052 USA
[4]School of Software and Microelectronics, Peking University, Beijing 102600 China
[5]School of Syber Engineering, Xidian University, Xi'an 710071 China

Corresponding author: Hong Li (e-mail: lihong@iie.ac.cn).

This work was supported in part by the Beijing Natural Science Foundation under Grant 4184085, in part by the National Natural Science Foundation of China under Grant 61702503, Grant 61602053 and Grant 61672415, in part by the Youth Science and Technology Innovation Foundation (North China University of Technology) under Grant 1473009, and in part by the Natural Science Foundation of Shaanxi Province under Grant 2017JM6054.

ABSTRACT In distributed peer-to-peer (P2P) applications, peers self-organize and cooperate to effectively complete certain tasks such as forwarding files, delivering messages, or uploading data. Nevertheless, users are selfish in nature and they may refuse to cooperate due to their concerns on energy and bandwidth consumption. Thus each user should receive a satisfying reward to compensate its resource consumption for cooperation. However, suitable incentive mechanisms that can meet the diverse requirements of users in dynamic and distributed P2P environments are still missing. On the other hand, we observe that Blcokchain is a decentralized secure digital ledger of economic transactions that can be programmed to record not just financial transactions and Blockchain-based cryptocurrencies get more and more market capitalization. Therefore in this paper, we propose a Blockchain based truthful incentive mechanism for distributed P2P applications that applies a cryptocurrency such as Bitcoin to incentivize users for cooperation. In this mechanism, users who help with a successful delivery get rewarded. As users and miners in the Blockchain P2P system may exhibit selfish actions or collude with each other, we propose a secure validation method and a pricing strategy, and integrate them into our incentive mechanism. Through a game theoretical analysis and evaluation study, we demonstrate the effectiveness and security strength of our proposed incentive mechanism.

INDEX TERMS Incentive mechanism, P2P applications, data transmissions, Bitcoin System, collusion attacks, pricing strategy.

I. INTRODUCTION

PEER-to-Peer (P2P) applications are featured by distributed architectures that partition tasks or work loads between peers without a trusted authority. Peers self-organize and collaborate in certain tasks such as forwarding files, delivering messages, or uploading data. Example P2P applications include mobile data offloading that allows mobile users to cooperatively deliver cellular network data by exploiting complementary network technologies (e.g., WiFi and femtocell) [1], Delay Tolerant Social Network (DTSN) [2] where users opportunistically forward messages for others that share common interests by following a store-carry-forward mechanism, and mobile crowdsensing in which smartphone users collaboratively upload data for the purpose of reducing energy consumption and mobile data cost [3].

The effectiveness of data transfer, packet forwarding, or data collection in the P2P applications mentioned above relies on the cooperation of mobile users. As data transmissions in such P2P applications often happen in an opportunistic manner, uncooperative behaviors would lead to low delivery success ratio and long delivery latency. On the other hand, autonomous mobile users may exhibit selfish behaviors, re-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2821705, IEEE Access

fusing to cooperate in data transmissions for the concerns on energy and bandwidth consumption. Therefore the participating users should be provided with enough rewards for cooperation. However, the properties of P2P applications bring challenges into the design of an incentive mechanism [4], [5]. First, most P2P applications exploit opportunistic connections among mobile users without knowing who will be the next hop and how many users will get involved; thus it is hard to know who will get rewarded and how many rewards shall be paid. Second, in a distributed environment, users may not believe that they can get their rewards as there is no pre-established trust between each other. Moreover, the uncontrolled environment may lead to some disorders in an incentive system. Finally, users may have different valuations on consumed resources and different requirements on the returns.

Many incentive mechanisms have been proposed and implemented, including the reputation systems, the Tit-for-Tat schemes, and the credit based approaches. A reputation system [6], [7] can help a user identify uncooperative peers by monitoring the behaviors of its neighbors during data transmissions and propagating the uncooperative reputation throughout the network. Such systems generally lack the considerations on a formal specification and analysis of the incentive types and on how to define the reputation of a new user. Tit-for-Tat schemes [8], [9] stimulate mobile users to cooperate by exchanging equal services among them based on what contributions they have done for others. Each user receives as much service as it has done for its neighbors based on its history behavior. These schemes are restricted to applications with long session durations that can provide many opportunities for reciprocation between pairs of users. Another challenge of Tit-for-Tat is its hardness of meeting the different service requirements of the users. Credit based approaches [10], [11] provide incentives by paying the cooperative users certain amount of credit or virtual money. Such approaches could be the most promising due to their explicit and flexible incentive methods; nevertheless, most credit based incentive schemes either rely on a central trusted authority or do not give an explicit digital currency system that is provably secure, leading to possible system collapses.

Blockchain is a decentralized digital ledger of economic transactions that can be programmed to record not just financial transactions, and it has the advantages of transparency, security and speed [12]–[14]. Blockchain-based cryptocurrencies have recently gained a noticeable popularity, and their current market capitalization is over $566 billion. The security of Blockchain depends on a majority of the computing power instead of a central authority, thus eliminating the risks of one taking control over the system, generating inflation, or completely shutting down the system. In this paper, we exploit Blockchain transactions to incentivize users to cooperate in P2P applications.

The basic idea of our incentive scheme is to employ Blockchain transactions to reward those intermediate nodes that contribute to a successful delivery from the sender to the receiver. If an intermediate node helps transmit the data, the next-hop node sends it a signed acknowledgement which is used as a proof of getting the rewards. The miners in the Blockchain P2P system are in charge of verifying whether there is a successful delivery from the sender to the receiver, and examining the validity of the signed acknowledgements provided by the cooperative intermediate nodes in a successful delivery. The Blockchain P2P system is an independent system from P2P applications, the transactions generated in P2P applications will send to the miners in the Blockchain P2P system for the verification. This brings another concern: if a miner can see the content of a signed acknowledgement, she can disguise as a cooperative intermediate node to get the payment. To overcome this problem, we extend the Blockchain transaction syntax to support a secure validation of the acknowledgement by using commutative encryptions. We also propose a pricing strategy to defend the possible attacks resulted from selfish users and to prevent their collusion.

The major contributions of the paper are summarized as follows:

- We design a Blockchain-based truthful incentive mechanism that can meet the diverse requirements of users in dynamic and distributed P2P environments.
- We introduce a secure validation method to keep the to-be-verified content secret from the miners in the Blockchian P2P system.
- We propose a pricing strategy to prevent selfish users from exhibiting selfish actions and to defend the collusion attacks resulted from them.
- We further employ a game theoretical analysis and simulation study to demonstrate the security and efficiency of our incentive mechanism.

The remainder of the paper is structured as follows. Section II outlines the related work. In Section III, we introduce the Blockchain P2P system and the basic cryptographic primitives employed in this paper. Our incentive scheme is detailed in Section IV, followed by a comprehensive security analysis and evaluation in Section V. The paper is concluded in Section VI.

II. RELATED WORK

The incentive schemes for P2P applications fall into three categories: Reputation, Barter (Tit-for-Tat), and Credit (virtual money).

A. REPUTATION-BASED APPROACHES

In a reputation system [6], [7], [15], [16], each user is given a score that can be interpreted as the probability of an entity behaving honestly. Such a system provides us with information regarding the honesty of the peers, therefore can be utilized to identify misbehaving users. For example, a watchdog is used by a user in [16] to monitor the behavior of each neighbor to make sure that the neighbor forwards others' data. If misbehaviors are detected, the user broadcasts the uncooperative reputation of the neighbor to other users in the network. Burak et al. [15] combined centralized reputation-based evaluation with collaborative reputation values based on votes. Reputation systems generally suffer from the following drawbacks: i) no formal specification and analysis of the types of the incentives is provided [10]; ii) the possibility of selfish users colluding with each other to maximize their welfare is generally ignored; and iii) the reputation-based approaches are known to be vulnerable to Sybil attacks [17] and whitewashing attacks [18].

B. TIT-FOR-TAT SCHEMES

Barter (Tit-for-Tat) based schemes [8], [9], [19] stimulate mobile users to cooperate by exchanging equal services based on what contributions they have done for others. In [8], Buttyan et al. proposed the use of pair-wise Tit-for-Tat (TFT) as an incentive mechanism for DTNs, in which each user estimates the contribution levels of its neighbors based on their behaviors in history, and then forwards as much traffic to its neighbors in accordance with their contribution levels. Parisa et al. [9] used the T-Chain incentive mechanism to discourage free-riding in video streaming applications. Tit-for-Tat schemes are restricted to applications with long session durations that can provide many opportunities for reciprocation between pairs of users [20]. Another challenge of Tit-for-Tat is its hardness to meet the different service requirements of the users.

C. CREDIT-BASED SCHEMES

In Credit based systems [10], [11], [21]–[24], a central authority assigns certain virtual money to each user. When a user needs others' help (for example, to forward a message), it should pay the helper certain amount of virtual money. As the amounts of payments rely on the reports of the users, a selfish user may attempt to cheat the system to maximize its expected welfare. For example, a selfish node may withhold its acknowledgement, or collude with other nodes to forge acknowledgements, if such actions can maximize its welfare. Some effort has been made to counter such cheating behaviors. Zhong et al. [10] proposed a cheat-proof, credit-based system for stimulating cooperation among selfish nodes in mobile ad hoc networks. Felegyhazi et al. [25] and Zhang et al. [11] developed game theory frameworks based on the use of pricing strategies. These schemes assume that a routing path between the sender and the receiver is determined before data transmission occurs. Without knowing who will be the next hop and how many users will get involved, it is hard to know who will get rewarded and how many rewards shall be paid.

Other related credit-based schemes include [22] and [21]. Zhu et al. [22] proposed a layered incentive scheme for dynamic routing in DTNs. In this scheme, the sender first generates a base-layer message containing the payment policy. Then each of the intermediate nodes who agrees with the payment policy generates a new layer based on the previous layer by appending a non-forged digital signature. After receiving all the collected layered messages, the last intermediate node sends them to a trusted center, i.e., a bank, for validation and payment assignment according to the payment policy. Obviously, this mechanism emphasizes the generation and verification of the secure layered messages but does not involve a detailed pricing strategy. Chen et al. [21] presented a pricing strategy running on top of a given DTN routing module, which works by setting the sender's payment and the intermediate nodes' rewards to ensure that selfish actions do not result in larger rewards. Bogliolo et al. [26] investigated the combined use of credit and reputation-based schemes to establish ad-hoc connections. We notice that all the credit based incentive schemes rely on central trusted authorities that do not exist in P2P applications. Furthermore, no explicit virtual digital currency system that is provably secure was proposed by any credit based system. Nevertheless, lacking a secure currency system could result in a collapsed incentive mechanism, just like the economy could break down if the banking system is not secure or is out of control.

Our incentive scheme is inspired by the existing research, but is very different. It is based on the Bitcoin system whose security is directly dependent on the majority of the computing power. We propose a secure validation approach carried out by the miners in the Bitcoin system instead of a trusted third authority, and a pricing strategy for the secure Bitcoin system to defend against selfish actions and prevent collusion attacks resulted from selfish users. Compared with the existing ones, our scheme does not rely on a trusted third authority but it can incentivize each node in P2P applications to honestly follow the designed protocol.

III. PRELIMINARIES AND THE THREAT MODEL

A. BLOCKCHAIN P2P SYSTEM

Since Bitcoin [27] came out in 2008, its underlying technique, blockchain has attracted lots of attentions from academia and industry. Many versions of Blockchain were released as open-source softwares [12], [27], [28]. Compared with other payment systems, Blockchain is peer-to-peer, i.e. users can transact directly without a trusted authority. Each transaction in Blockchain involves a sender A and a recipient B who are respectively identified by their public keys PK_A and PK_B, and miners who verify the correctness of the transaction.

A typical transaction works for the money transfer as follows: (1) A first creates a transaction $T_x = (T_y, PK_B, v, Sig_{SK_A}(T_y, PK_B, v))$ and broadcasts it to other nodes in the network, where T_y is a previous transaction, v is the amount of coins, SK_A is A's private key, and $Sig_{SK_A}(T_y, PK_B, v)$ is a signature on T_x signed by A; (2) miners who maintain a chain of blocks verify if the money from transaction T_x has not been spent by A and solve a hard mining problem; (3) a miner that completes the validation and the mining problem adds the transaction to a block and broadcasts it to other nodes. Blockchain also supports a more generalized transaction [29], [30], which considers the functionality money transfer, i.e., a sender conditionally transfers its money to a receiver. The workflow of the generalized transaction is similar to the typical one, but the context of the transaction is different. The context of a generalized transaction $T_x = (T_y, \pi_x, v, \sigma)$ is shown in Fig. 1, where the in-script σ refers to some information for validation provided by the sender of the transaction, and the output-script π_x with a Boolean output refers to the recipient of the transaction. A receiver redeeming T_x is valid only if π_x evaluates to true.

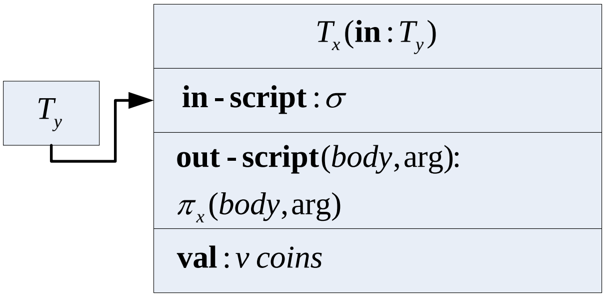
$T_x(\mathbf{in}:T_y)$
T_y
$\mathbf{in\text{-}script}:\sigma$
$\mathbf{out\text{-}script}(body, \mathrm{arg})$:
$\pi_x(body, \mathrm{arg})$
$\mathbf{val}: v\ coins$

FIGURE 1: A Bitcoin transaction.

Deposit protocols [31]–[33] have been implemented by exploiting the Blockchain P2P system, in which each user is required to initially put aside a certain amount of money, which will be paid back to the user once he/she honestly follows the protocol; otherwise the deposit is given to the other parties and compensates them for the fact that the protocol terminates prematurely. The design philosophy and features of Blockchain make it an attractive way to address the incentive challenges in peer-to-peer networks that lack trusted third parties.

B. COMMUTATIVE ENCRYPTION

In this work, we employ commutative encryption as one of the basic cryptographic primitives. A commutative encryption function [34] is a family of bijections $f : M \times K \to M$ such that for a given $m \in M$ we have

$$f_a(f_b(m)) = f_b(f_a(m))$$

for any $a, b \in K$. A commutative encryption enables a plaintext to be encrypted more than once using different users' public keys (say a and b). One cannot decrypt the ciphertext $f_b(f_a(m))$ without the help of the other user. The order of the keys used in encryption and decryption do not affect the computational result. Example commutative cryptosystems include Pohlig-Hellman [35] and Massey-Omura [36].

C. THREAT MODEL AND ASSUMPTIONS

Fig. 2 illustrates a typical architecture of P2P applications. The system consists of senders, intermediate nodes, and receivers. Senders transmit certain files, messages, or the sensed data to the receivers with the help of the intermediate nodes. The numbers of senders and receivers are different in different P2P applications. For example, there are 1 sender and n receivers in mobile data offloading, while in the context of DTN there are only 1 sender and 1 receiver. In this paper, we consider a simple case with 1 sender and 1 receiver. Our incentive scheme can be easily extended to the cases with 1 sender and multiple receivers, and then the more complex cases with multiple senders and receivers.

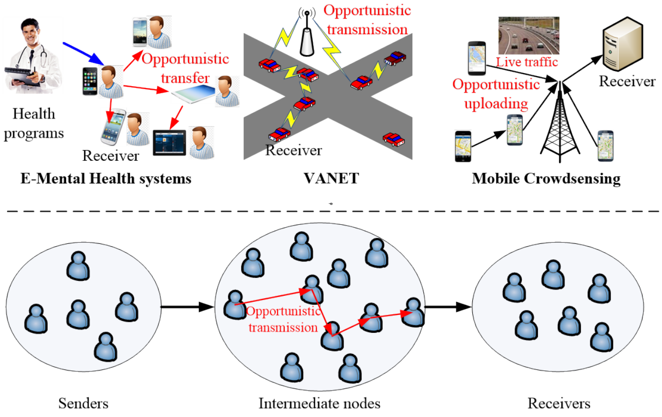

FIGURE 2: P2P application system.

Data transmissions in P2P applications rely on the cooperation between intermediate nodes. To incentivize the cooperations, senders give certain rewards to the nodes that help transmit the data. In this work, we assume that nodes are selfish but would take a rational decision to maximize their profit. Specially, each node may launch the following attacks:

- Refusing to Pay: A sender can refuse to pay back the intermediate nodes when the data are successfully delivered to the receiver.
- Denying Attack: The intermediate nodes or the receiver can deny that they have received the data from other nodes, which could prevent others from getting rewarded.
- Extending/Shortenning the Path: The intermediate node can extend or shorten the path to get more reward from the sender.
- Collusion Attack: Nodes can collude with each other to maximize their profit. In this work, we only consider the collusion among intermediate nodes or between an intermediate node and the receiver. We shall address the case where the sender colludes with the receiver in our future work by considering reputation based inventive systems.

IV. BLOCKCHAIN BASED TRUTHFUL INCENTIVE MECHANISM

In our model, we employ the idea of credit based incentives to motivate intermediate nodes to cooperate. In a credit based scheme, incentive can be considered as a transaction. When discussing a transaction, we should figure out the following questions: (1) who pays who; (2) how to pay the bill; (2) how much the payer should pay; and (4) how to guarantee that the transaction is secure.

A. WHO PAYS WHO?

When a sender wants to transmit a certain message to a receiver, there exist three different options to pay back the intermediate nodes. The first option is to let the receiver give rewards to all the intermediate nodes; but this approach allows malicious nodes to get high rewards by sending many fake messages. The second option is to give the rewards by both the sender and the receiver, which could suffer the same problem as the first option since the sender can collude with the intermediate nodes. The third option is for the sender to pay back the intermediate nodes when it figures out that the message is successfully delivered to the receiver, which is adopted by this work.

Another relevant question we need to answer is who should get the rewards. In this study, we choose to award only those nodes that contribute to a successful delivery, which means that an intermediate node cannot get a reward if the receiver does not receive the message correctly. To identify the intermediate nodes who help forward the message, the node in the next hop is required to send a signed acknowledgement back. Because a node is considered cooperative if and only if the node has a signed acknowledgement from its successor, it is important for an intermediate node to stimulate its successor by paying certain money to its successor for sending the signed acknowledgement.

B. HOW TO PAY THE BILL?

As mentioned before, intermediate nodes should be motivated to cooperate in a dynamic and distributed environment. In particular, a sender knows the receiver, but it does not know the route to the receiver. The sender should give rewards to the intermediate nodes who help transmit the message. Cooperative nodes can be divided into two types: *negative cooperative nodes* who help transmit the data but the receiver fails to receive the data, and *positive cooperative nodes* who help transmit the data and the receiver does successfully receive the data. In our consideration, the sender only pays back the positive cooperative nodes.

In our model, the sender employs the Blockchain P2P system to pay back the positive cooperative nodes. The workflow of the payment consists of three steps. In the first step, the sender publicizes a transmission task and makes a certain deposit that is used to pay back the positive cooperative nodes. In the second step, the sender transmits data to the receiver by opportunistic connections. In the last step, the positive cooperative nodes get their payments. For simplicity, we first discuss the case with only one positive cooperative node; then extend our scheme to the multiple positive cooperative node case.

1) One Positive Cooperative Node

Suppose that a sender A sends a message m to a receiver E. There is only one relay node B between the sender A and the receiver E. Let (PK_i, SK_i) be the public/private key pair of node i, $H()$ be a hash function, $E()$ be an encryption function, and $Sig()$ be a signature function. The workflow of the payment is elaborated as follows.

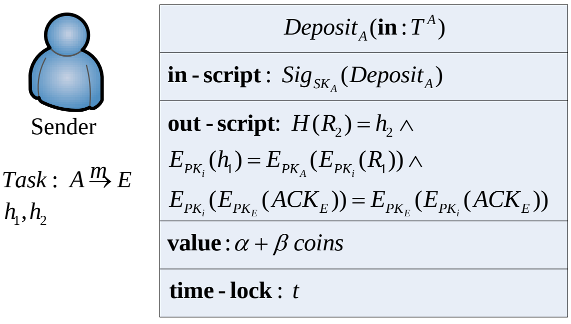
Sender
$Task:\ A \overset{m}{\rightarrow} E$
h_1, h_2
$Deposit_A(\mathbf{in}: T^A)$
in - script: $Sig_{SK_A}(Deposit_A)$
out - script: $H(R_2) = h_2 \wedge$
$E_{PK_i}(h_1) = E_{PK_A}(E_{PK_i}(R_1)) \wedge$
$E_{PK_i}(E_{PK_E}(ACK_E)) = E_{PK_E}(E_{PK_i}(ACK_E))$
value: $\alpha + \beta\ coins$
time - lock: t

FIGURE 3: A makes a deposit.

- Publishing a Transmission Task: When A wants to transmit data to E, it first generates two random numbers R_1 and R_2, where R_1 is used to prove that the node in the next hop receives the data correctly, and R_2 is used to prove that the receiver gets the data successfully. A keeps R_1 and R_2 secret, and publishes $h_1 = E_{PK_A}(R_1)$, $h_2 = H(R_2)$. Then A makes a deposit to commit that it will give rewards to B if B successfully transmits the data to the receiver E. A constructs a transaction $Deposit_A$ as shown in Fig. 3.

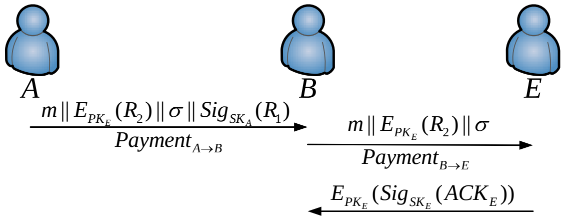
A
B
E
$m \| E_{PK_E}(R_2) \| \sigma \| Sig_{SK_A}(R_1)$
$Payment_{A \rightarrow B}$
$m \| E_{PK_E}(R_2) \| \sigma$
$Payment_{B \rightarrow E}$
$E_{PK_E}(Sig_{SK_E}(ACK_E))$

FIGURE 4: A transmits data to E.

- Data Transmission: As illustrated in Fig. 4, A first sends the message $m||E_{PK_E}(R_2)||\sigma||Sig_{SK_A}(R_1)$ to node B, where σ is a signature on $H(m)||E_{PK_E}(R_2)$. At the same time, A constructs a transaction $Payment_{A \rightarrow B}$ and broadcasts it to other nodes in the Blockchain P2P network for validation. Then, B sends the message $m||E_{PK_E}(R_2)||\sigma$ to the receiver E, constructs a transaction $Payment_{B \rightarrow E}$ and broadcasts it to the Blockchain P2P network. Finally, E receives the message, verifies the signature σ, and sends B a signed acknowledgement $E_{PK_E}(Sig_{SK_E}(ACK_E))$, which is encrypted by B's public key PK_B. Note that the transactions $Payment_{A \rightarrow B}$ and $Payment_{B \rightarrow E}$ represent a commitment that coins will be transferred only after the miners have verified them. The transactions in the data transmission are illustrated in Fig. 5.
- Obtaining the Payments: To redeem the rewards, B and E provide miners with $\{E_{PK_B}(ACK_E), E_{PK_B}(R_1)\}$ and $\{R_2, E_{PK_E}(ACK_E)\}$, respectively. Then, the miners validate the transactions. B and E can get the

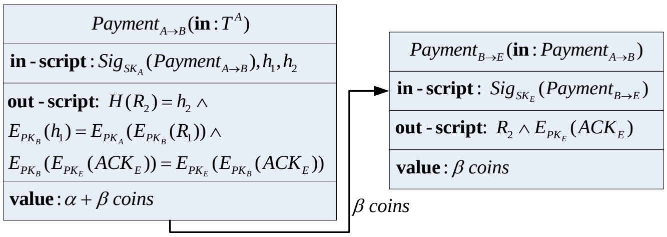
$Payment_{A\to B}(\mathbf{in}: T^A)$
$\mathbf{in\text{-}script}: Sig_{SK_A}(Payment_{A\to B}), h_1, h_2$
$\mathbf{out\text{-}script}: H(R_2)=h_2 \wedge$
$E_{PK_B}(h_1)=E_{PK_A}(E_{PK_B}(R_1)) \wedge$
$E_{PK_B}(E_{PK_E}(ACK_E))=E_{PK_E}(E_{PK_B}(ACK_E))$
$\mathbf{value}: \alpha+\beta\ coins$
$Payment_{B\to E}(\mathbf{in}: Payment_{A\to B})$
$\mathbf{in\text{-}script}: Sig_{SK_E}(Payment_{B\to E})$
$\mathbf{out\text{-}script}: R_2 \wedge E_{PK_E}(ACK_E)$
$\mathbf{value}: \beta\ coins$
$\beta\ coins$

FIGURE 5: Transactions in message transmission.

rewards if and only if the following conditions are satisfied:

(a) E can provide the random number R_2, which is verified by

$$H(R_2) = h_2;$$

(b) B can provide the random number R_1, which is verified by

$$E_{PK_B}(h_1) = E_{PK_A}(E_{PK_B}(R_1));$$

(c) B can provide the correct signed acknowledgement, which is verified by

$$E_{PK_B}(E_{PK_E}(ACK_E)) = E_{PK_E}(E_{PK_B}(ACK_E)).$$

2) Multiple Positive Cooperative Nodes

In this section, we extend our scheme to a more complex case where there are multiple positive cooperative nodes. Suppose that a sender A sends a message m to a receiver E, and B, C, D are the positive cooperative nodes who help A transmit the data to E. The workflow of the payment is elaborated as follows.

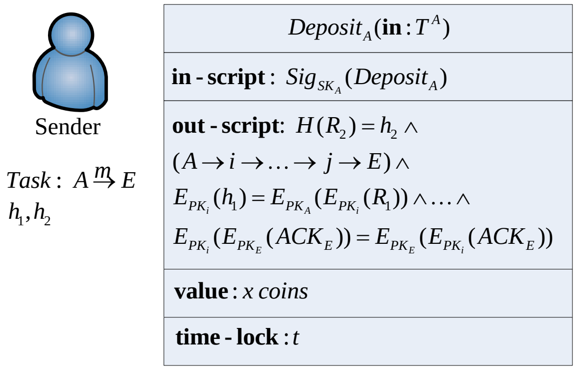
Sender
$Task: A \xrightarrow{m} E$
h_1, h_2
$Deposit_A(\mathbf{in}: T^A)$
$\mathbf{in\text{-}script}: Sig_{SK_A}(Deposit_A)$
$\mathbf{out\text{-}script}: H(R_2)=h_2 \wedge$
$(A \to i \to \ldots \to j \to E) \wedge$
$E_{PK_i}(h_1)=E_{PK_A}(E_{PK_i}(R_1)) \wedge \ldots \wedge$
$E_{PK_i}(E_{PK_E}(ACK_E))=E_{PK_E}(E_{PK_i}(ACK_E))$
$\mathbf{value}: x\ coins$
$\mathbf{time\text{-}lock}: t$

FIGURE 6: A publishes a task and makes a deposit.

- Publishing a Transmission Task: A announces a task $A \to E : m$ and generates two random numbers R_1 and R_2 that should be kept secret. Then A makes a deposit to commit that it will give the rewards to the positive cooperative nodes if the message is successfully delivered to the receiver E; otherwise A would get the deposit back. The transcript of the transaction is shown in Fig. 6.

- Data Transmission: The process of the data transmission from A to E is illustrated in Fig. 7. A first sends the message $m||E_{PK_E}(R_2)||\sigma||Sig_{SK_A}(R_1)$ to B, and constructs a transaction $Payment_{A\to B}$. Then, B, C, and D help A transmit the message $m||E_{PK_E}(R_2)||\sigma$ to the receiver E, and construct transactions $Payment_{B\to C}$, $Payment_{C\to D}$, and $Payment_{D\to E}$, respectively. C, D, and E send the signed encrypted acknowledgement back to B, C, and D, respectively. Note that the transactions $Payment_{A\to B}$, $Payment_{B\to C}$, $Payment_{C\to D}$, and $Payment_{D\to E}$ represent a commitment that coins will be transferred only after the miners have verified them. The Blockchain transactions in the data transmission are illustrated in Fig. 8.

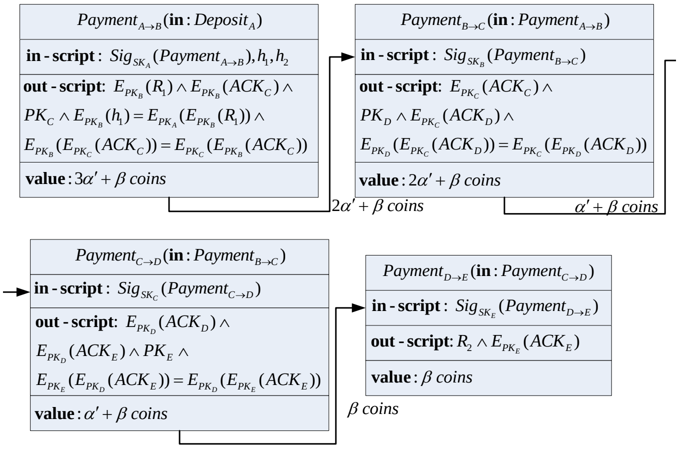
$Payment_{A\to B}(\mathbf{in}: Deposit_A)$
$\mathbf{in\text{-}script}: Sig_{SK_A}(Payment_{A\to B}), h_1, h_2$
$\mathbf{out\text{-}script}: E_{PK_B}(R_1) \wedge E_{PK_B}(ACK_C) \wedge$
$PK_C \wedge E_{PK_B}(h_1)=E_{PK_A}(E_{PK_B}(R_1)) \wedge$
$E_{PK_B}(E_{PK_C}(ACK_C))=E_{PK_C}(E_{PK_B}(ACK_C))$
$\mathbf{value}: 3\alpha'+\beta\ coins$
$Payment_{B\to C}(\mathbf{in}: Payment_{A\to B})$
$\mathbf{in\text{-}script}: Sig_{SK_B}(Payment_{B\to C})$
$\mathbf{out\text{-}script}: E_{PK_C}(ACK_C) \wedge$
$PK_D \wedge E_{PK_C}(ACK_D) \wedge$
$E_{PK_D}(E_{PK_C}(ACK_D))=E_{PK_C}(E_{PK_D}(ACK_D))$
$\mathbf{value}: 2\alpha'+\beta\ coins$
$2\alpha'+\beta\ coins$
$\alpha'+\beta\ coins$
$Payment_{C\to D}(\mathbf{in}: Payment_{B\to C})$
$\mathbf{in\text{-}script}: Sig_{SK_C}(Payment_{C\to D})$
$\mathbf{out\text{-}script}: E_{PK_D}(ACK_D) \wedge$
$E_{PK_D}(ACK_E) \wedge PK_E \wedge$
$E_{PK_E}(E_{PK_D}(ACK_E))=E_{PK_D}(E_{PK_E}(ACK_E))$
$\mathbf{value}: \alpha'+\beta\ coins$
$Payment_{D\to E}(\mathbf{in}: Payment_{C\to D})$
$\mathbf{in\text{-}script}: Sig_{SK_E}(Payment_{D\to E})$
$\mathbf{out\text{-}script}: R_2 \wedge E_{PK_E}(ACK_E)$
$\mathbf{value}: \beta\ coins$
$\beta\ coins$

FIGURE 8: Transactions in a multihop message transmission.

- Obtaining the Payments: After the data is successfully delivered to the receiver, all the positive cooperative nodes should get the rewards by providing the miners with the proofs that they did help transmit the data. To be more specific, B provides $\{E_{PK_B}(R_1), E_{PK_B}(ACK_C), PK_A, PK_C\}$; C provides $\{E_{PK_C}(ACK_C), E_{PK_C}(ACK_D), PK_D\}$; D provides $\{E_{PK_D}(ACK_D), E_{PK_D}(ACK_E), PK_E\}$; and E provides $\{R_2, E_{PK_E}(ACK_E)\}$. The transactions are considered to be valid if and only if the following conditions are satisfied:

  (a) E can provide the random number R_2, which is verified by

  $$H(R_2) = h_2;$$

  (b) There is a route from A to E, and the route can be determined by the transaction chain from A to E, as shown in Fig. 9.

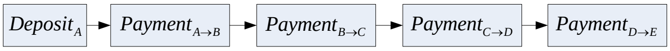

FIGURE 9: The transaction chain from A to E.

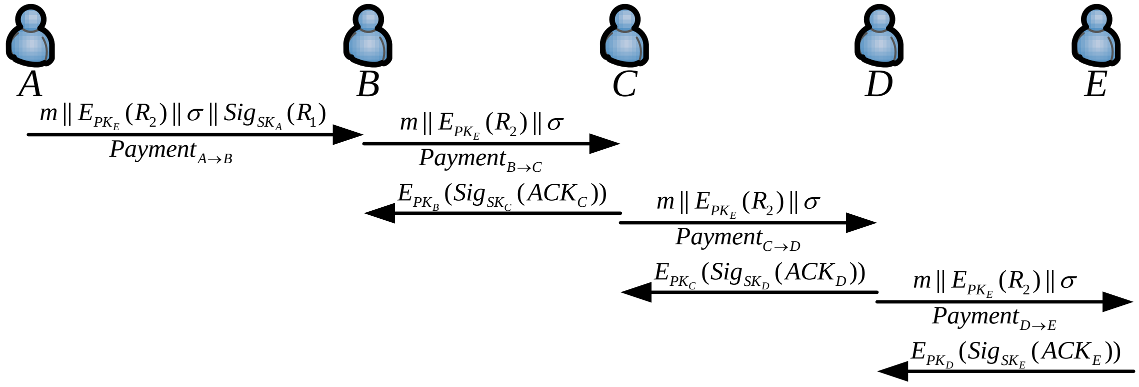
A
B
C
D
E
$m \,\|\, E_{PK_E}(R_2) \,\|\, \sigma \,\|\, Sig_{SK_A}(R_1)$
$Payment_{A\to B}$
$m \,\|\, E_{PK_E}(R_2) \,\|\, \sigma$
$Payment_{B\to C}$
$E_{PK_B}(Sig_{SK_C}(ACK_C))$
$m \,\|\, E_{PK_E}(R_2) \,\|\, \sigma$
$Payment_{C\to D}$
$E_{PK_C}(Sig_{SK_D}(ACK_D))$
$m \,\|\, E_{PK_E}(R_2) \,\|\, \sigma$
$Payment_{D\to E}$
$E_{PK_D}(Sig_{SK_E}(ACK_E))$

FIGURE 7: A transmits the data to E.

(c) B can provide the random number R_1, which can be verified by

$$E_{PK_B}(h_1) = E_{PK_A}(E_{PK_B}(R_1));$$

(d) B, C, D can provide the correct signed acknowledgements, which are verified by

$$\begin{aligned} E_{PK_B}(E_{PK_C}(ACK_C)) &= E_{PK_C}(E_{PK_B}(ACK_C)),\\ E_{PK_C}(E_{PK_D}(ACK_D)) &= E_{PK_D}(E_{PK_C}(ACK_D)),\\ E_{PK_D}(E_{PK_E}(ACK_E)) &= E_{PK_E}(E_{PK_D}(ACK_E)). \end{aligned}$$

C. HOW MUCH SHOULD THE PAYERS PAY?

A sender should determine the payment to the positive cooperative nodes for their help to transmit its data, and each positive cooperative node needs to determine the payment to its successor for sending the signed acknowledgement. Instead of considering the two components separately, we consider the final payment to the positive cooperative nodes and the receiver. Without loss of generality, we assume that A sends m to E via $P = (P_1, P_2, \cdots, P_n)$, the list of positive cooperative nodes who help the transmission. Then, the final payment to node i is computed by

$$p_i = \begin{cases} \alpha/2^{n-1}, & \text{if } i \in P,\\ \beta, & \text{if } i = E,\\ 0, & \text{otherwise.} \end{cases}$$

Note that in our implementation, A first makes a deposit; after determining the number of positive cooperative nodes, A determines the actual amount of coins given to them, and the residual deposit will return to A. For example, in the case of multiple positive cooperative nodes shown in Fig. 6, A first makes a deposit of $\alpha+\beta$ coins. After all the positive cooperative nodes have been identified (Fig. 8), A sets $\alpha'{=}\alpha/2^{3-1} = \alpha/4$.

D. IS IT SECURE?

1) Cheating Behaviors of the Participants

Since selfish mobile nodes always try to maximize their utilities, they may cheat. For example, they can act as a miner to get more information from the Blockchain transactions and then launch sophisticated attacks. In particular, a receiver may have one of the three selfish actions: *i) it does not send back a signed acknowledgement to its previous node or just simply sends the acknowledge to others; ii) it does not provide the validation information or provides a bogus validation information; and iii.a) it does not receive the data but falsely claims that it has received the data.* An intermediate node can have another cheating behavior except the first two selfish actions mentioned above: *iii.b) it does not help with transmitting the data but falsely claims that it has transmitted the data.*

Note that any of the selfish actions mentioned above can be further complicated by the collusion of two or more nodes. We assume that collusion only occurs between neighbors. This is a reasonable assumption in opportunistic transmissions because a node hardly gets in contact with a non-neighbor node. Moreover, we can add a lock-time into each payment transaction, which makes it more difficult for a node to contact its non-neighbors. We consider two kinds of collusion attacks:

- A receiver colludes with its neighbors. The receiver and its neighbors can forge a bogus path from the sender to the receiver. For example, suppose that the true path is $A \to B \to C \to D \to E$. After receiving the data from D, E does not send the signed acknowledgement to D, or it does not provide the validation information. Instead, E chooses some nodes from its neighbors to create a bogus path, i.e., $A \to E_1 \to E_2 \to E_3 \to E$.
- An intermediate node colludes with its neighbors. The intermediate node collude with its neighbors to extend or shorten the path. As shown in Fig. 10, B can choose some nodes from its neighbors to create a bogus path $A \to B \to B_1 \to B_2 \to B_3 \to C \to D \to E$, or B colludes with C to shorten the path to obtain a new path $A \to B \to D \to E$.

2) Data-Transmission Game.

To study the security of our incentive mechanism, we employ a static game to analyze the cooperative behaviors of the intermediate nodes. Through the Nash equilibrium results of the game, we can obtain the best strategies of the players

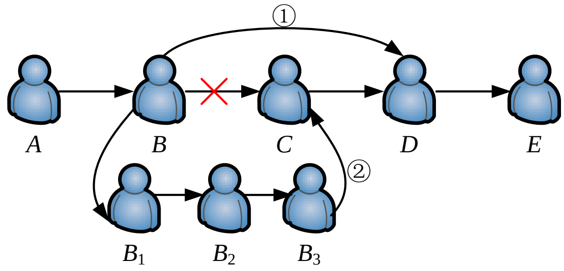

FIGURE 10: An intermediate node colludes with its neighbors.

under different pricing strategies. By setting a suitable pricing strategy, we can guarantee the security of our incentive mechanism against the selfish behaviors of the users and the collusion attacks. The model of the data-transmission game is described as follows.

Players. This game has $n+1$ players, the positive cooperative nodes $P = (P_1, P_2, \cdots, P_n)$ and the receiver E.

Strategies. Each player i has two possible actions: play honestly or play selfishly. If player i plays honestly, it follows the protocol; otherwise, it plays selfishly, either behaves selfishly itself or colludes with its neighbors. We denote the strategy of node i by s_i. Then s_i is either $Honest$ or $Selfish$.

Utilities. Player i can get its utility by deducting its cost from its received payment. Without colluding with its neighbors, the utility of u_i is computed by

$$u_i = \begin{cases} \alpha/2^{n-1} - c_i, & i \in P \text{ and } s_i = Honest, \\ \beta - c_E, & i = E \text{ and } s_i = Honest, \\ 0, & i \in P \text{ and } s_i = Selfish, \\ 0, & i = E \text{ and } s_i = Selfish. \end{cases}$$

where c_i is the cost of i for transmitting the data, sending a signed acknowledgement, and providing the validation information, c_E is the cost of the receiver E for sending a signed acknowledgement and providing the validation information.

When player i colludes with others, it is more complicated because the utility should consider the success probability of the collusion attack. We discuss the details in Section V. Here we present some definitions for the security analysis of our incentive scheme.

Definition 1. *The best response strategy for a player is a strategy that brings the maximum expected utility to itself, regardless of the strategies of all other players.*

To meet the security requirement, we need to design an incentive mechanism to discourage playing selfishly. In other words, we should make sure that $s_i = Honest$ is the best response strategy for each player in the game. We detail the following two cases: without and with collusion. More definitions about collusion resistance are given as follows.

Definition 2. *An incentive mechanism is receiver-collusion-resistant if the receiver and any group of its colluding neighbors cannot increase the expected sum of their utilities by using any strategy profile other than the one in which everybody plays honestly.*

Definition 3. *An incentive mechanism is intermediate-node-collusion-resistant, if any group of colluding intermediate nodes cannot increase the expected sum of their utilities by using any strategy profile other than the one in which everybody plays honestly.*

Definition 4. *An incentive mechanism is secure if $s_i = Honest$ is the best response strategy for each player and the game is receiver-collusion-resistant and intermediate-node-collusion-resistant.*

V. SECURITY ANALYSIS AND UTILITY EVALUATION

A. SECURITY ANALYSIS WITHOUT COLLUSION ATTACKS

Theorem 1. *In the data-transmission game, $s_i = Honest$ is the best response strategy for every player i if $\alpha > 2^{n-1}c_i$ and $\beta > c_E$.*

Proof. When player i plays honestly, we have

$$u_i = \begin{cases} \alpha/2^{n-1} - c_i, & i \in P, \\ \beta - c_E, & i = E. \end{cases}$$

E does not respond honestly (or an intermediate node P_i does not play honestly):

- E (or P_i) does not send the acknowledgement to its previous node P_n (P_{i-1}) or simply sends it to another node. If E (P_i) does not send $E_{PK_n}(Sig_{SK_E}(ACK_E))$ ($E_{PK_{i-1}}(Sig_{SK_i}(ACK_i))$) to P_n (P_{i-1}), P_n (P_{i-1}) will not provide the validation information; thus E (P_i) can not get the payment from the transaction $Payment_{P_n \to E}$ ($Payment_{P_{i-1} \to P_i}$). If E (P_i) sends the acknowledgement to another node, the node can not provide the validation information of P_n (P_{i-1}) because P_n (P_{i-1}) keeps its acknowledgement secret by encryption, i.e., $E_{PK_n}(ACK_n)$ ($E_{PK_{i-1}}(ACK_{i-1})$). Thus, E (P_i) can not get the payment.
- E (or P_i) does not provide validation information or provides a bogus validation information. If E (P_i) does not provide the validation information, E (P_i) can not get the payment from $Payment_{P_n \to E}$ ($Payment_{P_{i-1} \to P_i}$). If E (P_i) provides a bogus validation information, the transactions $Payment_{A \to P_1}$ ($Payment_{P_{i-2} \to P_{i-1}}$) and $Payment_{P_{n-1} \to P_n}$ ($Payment_{P_{i-1} \to P_i}$) can not be verified. Thus, E (P_i) can not get the payment.
- E (or P_i) does not receive the data but falsely claims that it has received the data. If E (P_i) does not receive the data, E can not provide R_1, then the transaction $Payment_{A \to P_1}$ can not be verified. Thus E (P_i) can not get the payment. In addition, if the data is important to E, cheating will damage its benefit.

As $\alpha > 2^{n-1}c_i$, we have $u'_E = 0 < \beta - c_E = u_E$ and $u'_i = 0 < \alpha/2^{n-1} - c_i = u_i$. Therefore, E's (P_i's) utility is reduced by playing selfishly. Therefore, if $\alpha > 2^{n-1}c_i$ and

$\beta > c_E$, $s_i = Honest$ is the best response strategy for the payer i. □

B. SECURITY ANALYSIS WITH COLLUSION ATTACKS

We first consider the case when E colludes with its neighbors; then we analyze the case when an intermediate node colludes with its neighbors.

Theorem 2. *Our incentive mechanism is receiver-collusion-resistant if $\alpha < \beta/q^2$, where q is the probability that two arbitrary nodes encounter each other.*

Proof. We first consider the case with one conspired node; then we extend to the case of multiple conspired nodes.

Case 1. Suppose $G = \{E, E_1\}$ is a collusion group. G forges a bogus path with one positive cooperative node, i.e., $A \to E_1 \to E$. Let $E(u_G)$ denote the expected sum of the utility of G. Our goal is to show that

$$E(u_G) \leq u_E.$$

If E_1 gets R_1, E and E_1 can get the payment from A, which means that E_1 has encountered both E and A (with a probability of q^2). The expected sum of the payment of G is $p_G = q^2(\alpha + \beta) + (1 - q^2)\beta = q^2\alpha + \beta$. Considering the cost of E_1 to provide the validation information and to communicate with E, we have the expected sum of the utility of G to be $u_G = q^2\alpha + \beta - \beta - c_E = q^2\alpha - c_E$. Thus we obtain $u_G = q^2\alpha - c_E < \beta - c_E = u_E$.

Case 2. Suppose $G = \{E, E_1, \cdots, E_n\}$ is a collusion group. G forges a bogus path with multiple positive cooperative nodes, i.e., $A \to E_1 \to \cdots \to E_n \to E$. Let $E(u_G)$ denote the expected sum of the utility of G. Our goal is to show that

$$E(u_G) \leq u_E.$$

When $(A, E_1), (E_1, E2), \cdots, (E_n, E)$ encounter each other, G gets the payment. The expected sum of the payment of G is

$$\begin{aligned} p_G &= q^{n+1}(n\alpha/2^{n-1} + \beta) + (1 - q^{n+1})\beta \\ &= q^{n+1}n\alpha/2^{n-1} + \beta. \end{aligned}$$

Deducting the cost of G, we have the expected sum of the utility of G:

$$u_G = q^{n+1}n\alpha/2^{n-1} + \beta - n\beta - c_E.$$

As $\alpha < \beta/q^2$, we have

$$\begin{aligned} u_G &= q^{n+1}n\alpha/2^{n-1} + \beta - n\beta - c_E \\ &< \frac{q^{n+1}n\beta}{2^{n-1}q^2} - n\beta + \beta - c_E \\ &= (q^{n-1}/2^{n-1} - 1)n\beta + \beta - c_E \\ &< \beta - c_E = u_E. \end{aligned}$$

Therefore, if $\alpha < \beta/q^2$, our incentive mechanism is receiver-collusion-resistant. □

Theorem 3. *Our incentive mechanism is intermediate-node-collusion-resistant.*

Proof. An intermediate node can collude with its neighbors to extend or shorten the path.

Case 1. An intermediate node colludes with its neighbors to extend the path. We first Consider the case with one positive cooperative node $A \to B \to E$. Let $G = \{B, B_1, \cdots, B_n\}$ be the collusion group. G extends the path to $A \to B \to B_1 \to \cdots \to B_n \to E$. Let $E(u_G)$ denote the expected sum of utility of G. Our goal is to show that

$$E(u_G) \leq u_B,$$

where u_B is the utility of B to play honestly. As B indeed helped A transmit data to E, it can get all the needed validation information from A and E, which means that B can always launch a successful collusion attack. According to our pricing scheme, we have

$$\begin{aligned} E(u_G) &= (n+1)\alpha/2^n - c_B; \\ u_B &= \alpha - c_B. \end{aligned}$$

Let $f(x) = (x+1)/2^x,\ x \geq 1$. We have $f'(x) = 2^{-x}(1 - (1+x)x) < 0$. Thus, $f(x)$ is a monotonically decreasing function. Accordingly we have $f(n) < f(n-1) < \ldots < f(1)$. It is easy to see that

$$E(u_G) = (n+1)\alpha/2^n - c_B < \alpha - c_B = u_B.$$

Now we consider the case with multiple positive cooperative nodes. Let $u_B = \alpha' - c_B$. We can deduce that

$$E(u_G) = (n+1)\alpha'/2^n - c_B < \alpha' - c_B = u_B.$$

Case 2. An intermediate node colludes with its neighbors to shorten the path $A \to P_1 \to \cdots \to P_i \to P_{i+1} \to \cdots \to P_n \to E$. Let $G = \{P_i, P_{i+1}\}$ be a collusion group. Then G shortens the path to $A \to P_1 \to \cdots \to P_i \to P_{i+2} \to \cdots \to P_n \to E$. Let $E(u_G)$ denote the expected sum of the utility of G. Our goal is to show that

$$E(u_G) \leq u_{P_i} + u_{P_{i+1}},$$

where u_{P_i} and $u_{P_{i+1}}$ are respectively the utilities of P_i and P_{i+1} to play honestly. As P_i and P_{i+1} indeed helped transmit the data, they can get all the needed validation information. Thus they can launch a successful collusion attack. According to our pricing scheme, we have

$$\begin{aligned} E(u_G) &= \alpha/2^{n-2} - c_{P_i} - c_{P_{i+1}}; \\ u_{P_i} &= \alpha/2^{n-1} - c_{P_i}; \\ u_{P_{i+1}} &= \alpha/2^{n-1} - c_{P_{i+1}}. \end{aligned}$$

It is easy to see that

$$E(u_G) = u_{P_i} + u_{P_{i+1}}.$$

Therefore, our incentive mechanism is intermediate-node-collusion-resistant. □

The three theorems together prove the following theorem.

Theorem 4. *Our incentive mechanism is secure if $\alpha > 2^{n-1}c_{\max}$, $\beta > c_E$, and $\alpha < \beta/q^2$.*

C. EVALUATION

1) Overhead

We employ a laptop computer with an Intel Core i7-2640M Processor (4MB Cache, up to 2.8GHz) to implement a prototype of our system using the Crypto++5.62 library and consider a path of 5 hops, to evaluate the overhead of our incentive mechanism. The OS of the laptop is Windows 10 Pro 64. The length of a message payload is 1024 bytes, and the message digest function is MD-5. We consider three commutative encryption schemes: ElGamal with a modulus of 1024 bits, RSA with a modulus of 1024 bits, and RSA with a modulus of 3072 bits.

TABLE 1: The CPU processing time

	sender (ms)	intermediate nodes (ms)	receiver (ms)	miner (ms)
ElGamal 1024	28	17	13	17
RSA 1024	11	5	4	7
RSA 3072	63	39	21	12

We first evaluate the CPU processing time. In our incentive system, the major processing overhead is the R2 encryption operation, the message and R1 signing operations, and the transaction generating operation by the sender, the ACK signing and encryption operation (or the R1 decryption operation) and the transaction generating operation by each intermediate node, the message verification operation, the R2 decryption operation, and the ACK signing and encryption operation by the receiver, and the verification operation by the miners. The columns of Table I report the CPU processing time of the sender, an intermediate node (average), the receiver, and a miner. We observe that RSA has a much smaller overhead. Therefore if reducing overhead is the major objective, RSA is a better implementation choice. Compared with the scheme proposed by Zhu et al. [22], the CPU processing time of the sender in our approach is slightly larger, the average CPU processing time of the intermediate node is slightly smaller, and the CPU processing time of the receiver is smaller.

We next evaluate the bandwidth and storage requirements. Compared with the opportunistic routing protocols introduced in [37], [38] but without any incentive mechanism, the major increased message overhead includes the encrypted R2, the signed R1, and the signed and encrypted ACK. For ElGamal and RSA with a modulus of 1024 bits, the encrypted R2 takes about 128 bytes, the signed R1 takes about 128 bytes, the signed and encrypted ACK takes about 128 bytes; for RSA 3074 bits, the encrypted R2 takes about 384 bytes, the signed R1 takes about 384 bytes, and the signed and encrypted ACK takes about 384 bytes. The storage requirement for the Blockchain transactions is analyzed as follows. For RSA 1024 and ElGamal 1024, each transaction requires at least 1 byte for the previous transaction reference, 128 bytes for the in-script, 1 byte for the Bitcoin value, and 128 bytes for out-script; adding up together we get 258 bytes for a minimum-sized Bitcoin transaction. For RSA 3074, each transaction requires 384 bytes for the in-script and 384 byes for the out-script, resulting in a 770-byte minimum-sized Bicoin transaction. Note that the process of miners verifying the transactions does not affect the cost of the routing protocols, and the throughput of processing the transaction is determined by Blockchain consensus algorithm [12].

2) Utility Evaluation

In this section we evaluate the utilities of the players under different strategies. We set $\alpha = 16 \times 10^{-3}$ coins, $\beta = 0.5 \times 10^{-3}$ coins, $c_E = 0.1 \times 10^{-3}$ coins, and $c_{P_i} = 1 \times 10^{-3}$ coins. The number of positive cooperative nodes is varied from 1 to 10 with a step size of 1, and the encounter probability is increased from 0 to 0.5 with a step size of 0.05.

Utility of a positive cooperative node: Fig. 11 shows the impact of the number of positive cooperative nodes and the playing strategies on the utility of a positive cooperative node. We observe that the node's utility indeed demonstrates diminishing returns when the number of positive cooperative nodes increases. Extending the length of the path ($k = 3, k = 5$) leads to the drop of P_i's utility. When $\alpha > 2^{n-1} c_{P_i}$, i.e., $n < 5$, playing honestly is the best response strategy for P_i because the utility by playing honestly is larger than that by playing selfishly. When $n >= 5$, the P_i's utility is below zero, which is hard to happen as rational participants want to gain benefits.

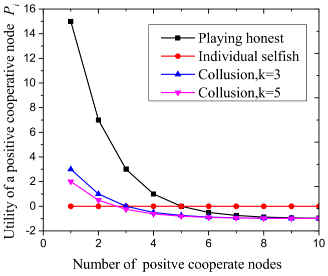

FIGURE 11: Utility of a positive cooperative node.

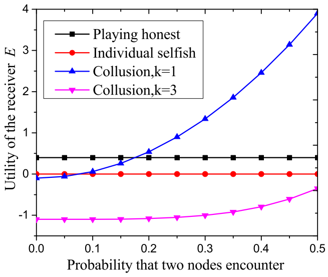

FIGURE 12: Utility of the receiver.

Utility of the receiver: Fig. 12 shows the impact of the encounter probability and the playing strategies on the utility of E. We observe that the success rate of collusions is higher when the encounter probability is higher. The utility obtained when E colluding with $k = 1$ intermediate node is larger than that with multiply intermediate nodes, i.e., $k = 3$. When $\alpha < \beta / q^2$, i.e., $q < 0.177$, the utility by playing honestly is larger than that by playing selfishly, which is in accordance with our theoretical analysis results.

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, we propose a Blockchain based incentive mechanism that can meet the diverse requirements in a dynamic and distributed P2P environment. In our incentive mechanism, intermediate nodes who contribute to a successful delivery can obtain rewards from Blockchain transactions. The transactions are verified by the miners in a secure way by using commutative encryptions. A pricing strategy is proposed to guarantee the security of our incentive mechanism. We also employ a static game model to demonstrate the security strength of our incentive mechanism.

In our future research, we will consider the issue of a sender colluding with its receiver. One possible solution is to introduce reputation into our incentive scheme. In this reputation based incentive scheme, the collusion group of the sender and the receiver will be put into a blacklist by the accusation of the intermediate nodes. We also will consider the contradiction of incentive and privacy in our scheme. We will bring certain cryptographic extensions such as zero-knowledge proof and blind signature to Blockchain for achieving fully anonymous currency transactions.

REFERENCES

[1] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" IEEE/ACM Transactions on Networking, vol. 21, no. 2, pp. 536–550, 2013.

[2] C. E. Baker, A. Starke, T. G. Hill-Jarrett, and J. McNair, "In vivo evaluation of the secure opportunistic schemes middleware using a delay tolerant social network," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 2537–2542.

[3] W. Sherchan, P. P. Jayaraman, S. Krishnaswamy, A. Zaslavsky, S. Loke, and A. Sinha, "Using on-the-move mining for mobile crowdsensing," in Mobile Data Management (MDM), IEEE 13th International Conference on. IEEE, 2012, pp. 115–124.

[4] F. L. Haddi and M. BenchaÃŕba, "A survey of incentive mechanisms in static and mobile p2p systems," Journal of Network and Computer Applications, vol. 58, pp. 108 – 118, 2015.

[5] Y. He, M. Chen, B. Ge, and M. Guizani, "On wifi offloading in heterogeneous networks: Various incentives and trade-off strategies," IEEE Communications Surveys Tutorials, vol. 18, no. 4, pp. 2345–2385, 2016.

[6] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," Computer Networks, vol. 90, pp. 49–73, 2015.

[7] P. Maryam, K. Burak, S. Tolga, F. Luca, and S. Houbing, "Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing," IEEE Access, vol. 5, pp. 1382–1397, 2017.

[8] L. Buttyan, L. Dora, M. Félegyházi, and I. Vajda, "Barter-based cooperation in delay-tolerant personal wireless networks," in World of Wireless, Mobile and Multimedia Networks(WoWMoM). IEEE International Symposium on. IEEE, 2007, pp. 1–6.

[9] R. Parisa, J.-W. Carlee, S. Kyuyong, I. Youngbin, L. Jongdeog, and H. Sangtae, "Svc-tchain: Incentivizing good behavior in layered p2p video streaming," in IEEE Conference on Computer Communications(INFOCOM 2017). IEEE, 2017, pp. 1–9.

[10] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in INFOCOM. IEEE, 2003, pp. 1987–1997.

[11] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun, "Incentive mechanism for proximity-based mobile crowd service systems," in INFOCOM. IEEE, 2016, pp. 1–9.

[12] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," Future Generation Computer Systems, 2017.

[13] R. Khalil and A. Gervais, "Revive: Rebalancing off-blockchain payment networks," in CCS, 2017.

[14] S. Matsumoto and R. M. Reischuk, "Ikp: Turning a pki around with decentralized automated incentives," in 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 410–426.

[15] K. Burak, P. M. Glasser, and F. Luca, "Crowdsensing with social network-aided collaborative trust scores," in IEEE Global Communications Conference (GLOBECOM). IEEE, 2015, pp. 1–6.

[16] Q. He, D. Wu, and P. Khosla, "Sori: a secure and objective reputation-based incentive scheme for ad-hoc networks," in IEEE WCNC. IEEE, 2004, pp. 825–830.

[17] B. N. Levine, C. Shields, and N. B. Margolin, "A survey of solutions to the sybil attack," University of Massachusetts Amherst, Amherst, MA, 2006.

[18] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," Selected Areas in Communications, IEEE Journal on, vol. 24, no. 5, pp. 1010–1019, 2006.

[19] Y. Lee, S. Kang, C. Min, Y. Ju, I. Hwang, and J. Song, "CoMon+: A cooperative context monitoring system for multi-device personal sensing environments," IEEE Transactions on Mobile Computing 2016, vol. 15, no. 8, pp. 1908–1924, 2016.

[20] T. Ning, Z. Yang, X. Xie, and H. Wu, "Incentive-aware data dissemination in delay-tolerant mobile networks," in Sensor, Mesh and Ad Hoc Communications and Networks (SECON'11), 8th Annual IEEE Communications Society Conference on. IEEE, 2011, pp. 539–547.

[21] B. Chen and M. C. Chan, "Mobicent: a credit-based incentive system for disruption tolerant network," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–9.

[22] H. Zhu, X. Lin, R. Lu, and X. S. Shen, "A secure incentive scheme for delay tolerant networks," in Communications and Networking in China (ChinaCom). Third International Conference on. IEEE, 2008, pp. 23–28.

[23] W. Li, X. Cheng, R. Bie, and F. Zhao, "An extensible and flexible truthful auction framework for heterogeneous spectrum markets," IEEE Trans. on Cognitive Commun. Network., vol. 2, no. 4, pp. 427 – 441, 2016.

[24] Y. Wang, Z. Cai, G. Yin, and Y. Gao, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," Comput. Netw., pp. 157 – 171, 2016.

[25] M. Felegyhazi, J.-P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," Mobile Computing, IEEE Transactions on, vol. 5, no. 5, pp. 463–476, 2006.

[26] A. Bogliolo, P. Polidori, A. Aldini, W. Moreira, P. Mendes, M. Yildiz, C. Ballester, and J. M. Seigneur, "Virtual currency and reputation-based cooperation incentives in user-centric networks," in the 8th International Wireless Communications and Mobile Computing Conference (IWCMC), 2012, pp. 895–900.

[27] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Consulted, vol. 1, no. 2012, pp. 1–28, 2008.

[28] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2p mixing and unlinkable bitcoin transactions," in NDSS, 2017.

[29] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in CCS, 2015, pp. 706–719.

[30] R. Kumaresan, T. Moran, and I. Bentov, "How to use bitcoin to play decentralized poker," in CCS, 2015, pp. 195–206.

[31] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014, pp. 443–458.

[32] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "Fair two-party computations via bitcoin deposits," in Financial Cryptography and Data Security. Springer, 2014, pp. 105–121.

[33] I. Bentov and R. Kumaresan, "How to use bitcoin to design fair protocols," in Advances in Cryptology (CRYPTO). Springer, 2014, pp. 421–439.

[34] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 17, no. 6, pp. 774–778, 2007.

[35] S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.)," Information Theory, IEEE Transactions on, vol. 24, no. 1, pp. 106–110, 1978.

[36] J. Massey and J. Omura, "A new multiplicative algorithm over finite fields and its applicability in public key cryptography," EUROCRYPT'83, 1983.

[37] A. Boukerche and A. Darehshoorzadeh, "Opportunistic routing in wireless networks: Models, algorithms, and classifications," ACM Computing Surverys, vol. 47, no. 2, pp. 1–36, 2015.

[38] S. Batabyal and P. Bhaumik, "Mobility models, traces and impact of mobility on opportunistic routing algorithms: A survey," Communications Surveys Tutorials, IEEE, vol. 17, no. 3, pp. 1679–1707, 2015.