A Bitcoin Based Incentive Mechanism for Distributed P2P Applications

Yunhua He[1,2], Hong Li[2,3(✉)], Xiuzhen Cheng[2], Yan Liu[5], and Limin Sun[3,4]

[1] The School of Computer Science,
North China University of Technology, Beijing, China
heyunhua@ncut.edu.cn
[2] The Department of Computer Science,
George Washington University, Washington, DC, USA
lihong@iie.ac.cn
[3] Beijing Key Laboratory of IOT Information Security Technology,
IIE, CAS, Beijing, China
[4] University of Chinese Academy of Sciences, Beijing, China
[5] The School of Software and Microelectronics, Peking University, Beijing, China

Abstract. The effectiveness of distributed Peer-to-Peer (P2P) applications heavily relies on the cooperation of mobile users. 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. Therefore in this paper, we propose a Bitcoin based incentive mechanism for distributed P2P applications that applies the basic idea of Bitcoin to incentivize users for cooperation. In this mechanism, users who help with a successful delivery get rewarded. Through a game theoretical analysis and evaluation study, we demonstrate the effectiveness and security strength of our proposed incentive mechanism.

1 Introduction

Peer-to-Peer (P2P) applications [7,13] are featured by *distributed architectures* that partition tasks or work loads between peers without a trusted authority. Example P2P applications include mobile data offloading that allows mobile users to cooperatively deliver cellular network data by exploiting complementary network technologies (WiFi, femtocell, etc.), delay-tolerant networking where nodes opportunistically forward messages for others by following a store-carry-forward mechanism, and mobile crowdsensing in which users collaboratively upload data for the purpose of reducing energy consumption and mobile data cost.

The effectiveness of data transfer, packet forwarding, or data collection in the P2P applications relies on the cooperation of mobile users. Selfish users may be reluctant to cooperate in data transmissions for the concerns on energy and bandwidth consumption. Thus, they should be provided with enough rewards for cooperation. Many incentive mechanisms have been proposed and implemented, including the reputation systems, Tit-for-Tat schemes, and credit based

L. Ma et al. (Eds.): WASA 2017, LNCS 10251, pp. 457–468, 2017.
DOI: 10.1007/978-3-319-60033-8_40

approaches. Reputation systems [16,19] can help identify uncooperative users by computing users' reputation scores, but such systems generally lack the considerations on collusion attacks and on how to define the reputation of a new user. Tit-for-Tat schemes [15] stimulate mobile users to cooperate by exchanging equal services among them, but these schemes are restricted to applications with long session durations. Credit based approaches [4–6] 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.

Bitcoin is a decentralized digital currency that is provably secure. It has recently gained a noticeable popularity, and its current market capitalization is over $16 billion. The security of Bitcoin depends on a majority of the computing power instead of a central authority [17], thus eliminating the risks of one taking control over the system, generating inflation, or completely shutting down the system. In this paper, we exploit Bitcoin transactions to incentivize users to cooperate in P2P applications.

The basic idea of our incentive scheme is to employ Bitcoin 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 Bitcoin system are in charge of verifying whether there is a successful delivery, and examining the validity of the signed acknowledgements. 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 Bitcoin transaction syntax [8,14] to support a secure validation of the acknowledgement by using commutative encryptions [12]. We also propose a pricing strategy to defend the possible attacks resulted from selfish users and to prevent their collusions. The major contributions of the paper are summarized as follows:

- We design a Bitcoin-based 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 Bitcoin system, and 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 2 outlines the related work. In Sect. 3, we introduce the threat model and assumptions employed in this paper. Our incentive scheme is detailed in Sect. 4, followed by a comprehensive security analysis and evaluation in Sect. 5. The paper is concluded in Sect. 6.

2 Related Work

The incentive schemes for P2P applications fall into three categories: Reputation, Tit-for-Tat, and Credit. In a reputation system [16], each user is given a score interpreted as the probability of an entity behaving honestly, such a system can be utilized to identify misbehaving users. Reputation systems generally suffer from the following drawbacks: (i) the possibility of selfish users colluding with each other to maximize their welfare is generally ignored; and (ii) they are known to be vulnerable to Sybil attacks [9] and whitewashing attacks [21].

Tit-for-Tat based schemes [15] stimulate mobile users to cooperate by exchanging equal services based on what contributions they have done for others. Tit-for-Tat schemes are restricted to applications with long session durations that can provide many opportunities for reciprocation between pairs of users [18]. Another challenge of Tit-for-Tat is its hardness to meet the different service requirements of the users.

In Credit based systems [10,11,20,21], 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. Zhong *et al.* [21] proposed a cheat-proof, credit-based system for stimulating cooperation among selfish nodes in mobile ad hoc networks. The scheme assumes that a routing path between the sender and the receiver is determined before data transmission occurs. Zhu *et al.* [22] proposed a layered incentive scheme for dynamic routing in DTNs. This mechanism emphasizes the generation and verification of the secure layered messages but does not involve a detailed pricing strategy. Chen and Chan [4] presented a pricing strategy running on top of a given DTN routing module. 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.

3 Threat Model and Assumptions

A typical architecture of P2P applications 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 more complex cases with multiple senders and receivers.

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.

4 Bitcoin Based 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; and (3) how much the payer should pay.

4.1 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.

4.2 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 Bitcoin 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. 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.

(1) Publishing a Transmission Task: A announces a task $A \rightarrow E : m$ 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; otherwise A would get the deposit back. The transcript of the transaction [1,3] is shown in Fig. 1.

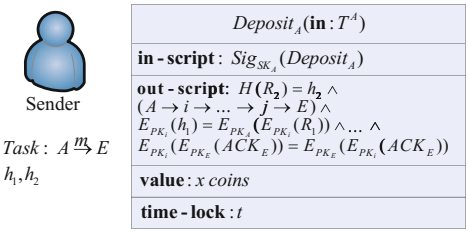

Fig. 1. A publishes a task and makes a deposit.

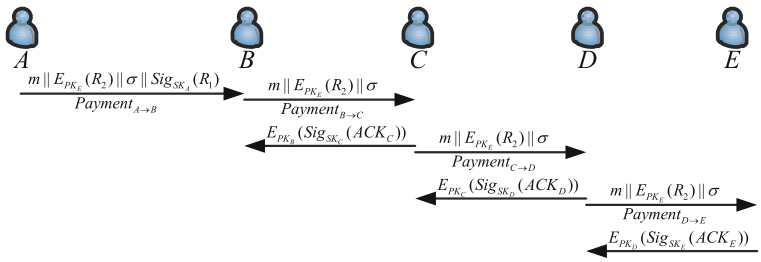

Fig. 2. A transmits the data to E.

(2) Data Transmission: The process of the data transmission from A to E is illustrated in Fig. 2. 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. The Bitcoin transactions in the data transmission are illustrated in Fig. 3.

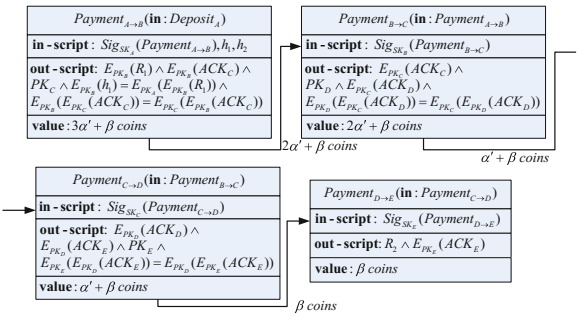

Fig. 3. Transactions in a multihop message transmission.

(3) 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. Specifically, 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:

- E can provide the random number R_2, which is verified by $H(R_2) = h_2$;
- There is a route from A to E, and the route can be determined by the transaction chain from A to E.
- B can provide the random number R_1, which can be verified by

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

- B, C, D can provide the correct acknowledgements, which are verified by

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

4.3 How Much Should the Payers Pay?

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. To be more specific, 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 can be computed by

$$p_i = \begin{cases} \alpha/2^{n-1}, & \text{if } i \in P, \\ \beta, & \text{if } i = E, \\ 0, & \text{otherwise.} \end{cases} \quad and \begin{cases} \alpha > 2^{n-1} c_{\max}, \\ \beta > c_E, \\ \alpha < \beta/q^2. \end{cases}$$

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. For example, in the case of multiple positive cooperative nodes shown in Fig. 1, A first makes a deposit of $\alpha + \beta$ coins. After all the positive cooperative nodes have been identified (Fig. 3), A sets $\alpha' = \alpha/2^{3-1} = \alpha/4$.

5 Security Analysis and Performance Evaluation

5.1 Data-Transmission Game Analysis Model

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 under different pricing strategies. The model of the data-transmission game analysis 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. Here we present some definitions for the security analysis of our incentive scheme.

Definition 1. *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 2. *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 3. *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.*

5.2 Security Analysis Without Collusion Attacks

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

Proof. When player i plays honestly, we have

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

If player i does not respond honestly, we have $u_i' = 0$ in this case. As $\alpha > 2^{n-1}c_i$, $u_i' = 0 < \alpha/2^{n-1} - c_i = u_i$; thus 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.

5.3 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., $A \rightarrow E_1 \rightarrow E$. 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* $\alpha < \beta/q^2$.

5.4 Performance Evaluation

We employ a laptop computer with an Intel Core i7-2640M Processor 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. CPU processing time

Commutative encryption	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

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 1 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.

Bandwidth and Storage. Compared with the opportunistic routing protocols introduced in [2] 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 Bitcoin 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.

6 Conclusion

In this paper, we propose a Bitcoin 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 Bitcoin 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.

Acknowledgment. This work was supported by the National Natural Science Foundations of China (61472418, 61672415, U1636120), the National Defense Science and Technology Innovation Foundation of the Chinese Academy of Sciences (CXJJ-16Z234) and the Science and Technology Innovation Service Capacity Building Project (PXM2017-014212-000002).

References

1. Andrychowicz, M., Dziembowski, S., Malinowski, D., Mazurek, L.: Secure multiparty computations on bitcoin. In: S&P, pp. 443–458. IEEE (2014)

2. Batabyal, S., Bhaumik, P.: Mobility models, traces and impact of mobility on opportunistic routing algorithms: a survey. IEEE Commun. Surv. Tuts. **17**(3), 1679–1707 (2015)
3. Bentov, I., Kumaresan, R.: How to use bitcoin to design fair protocols. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8617, pp. 421–439. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44381-1_24
4. Chen, B., Chan, M.C.: Mobicent: a credit-based incentive system for disruption tolerant network. In: INFOCOM, pp. 1–9. IEEE (2010)
5. Duan, Z., Li, W., Cai, Z.: Distributed auctions for task assignment and scheduling in mobile crowdsensing systems. In: ICDCS (2017)
6. Duan, Z., Yan, M., Cai, Z., Wang, X.: Truthful incentive mechanisms for social cost minimization in mobile crowdsourcing systems. Sensors **16**(4), 1–14 (2016)
7. Han, B., Hui, P., Kumar, V.A., Marathe, M.V., Shao, J., Srinivasan, A.: Mobile data offloading through opportunistic communications and social participation. IEEE Trans. Mob. Comput. **11**(5), 821–834 (2012)
8. Kumaresan, R., Moran, T., Bentov, I.: How to use bitcoin to play decentralized poker. In: CCS, pp. 195–206 (2015)
9. Li, H., He, Y., Cheng, X., Zhu, H., Sun, L.: Security and privacy in localization for underwater sensor networks. IEEE Commun. Mag. **53**(11), 56–62 (2015)
10. Li, W., Cheng, X., Bie, R., Zhao, F.: An extensible and flexible truthful auction framework for heterogeneous spectrum markets. IEEE Trans. Cogn. Commun. Netw. **2**(4), 427–441 (2016)
11. Li, W., Wang, S., Cheng, X.: Truthful multi-attribute auction with discriminatory pricing in cognitive radio networks. In: ACM MobiCom Workshop CRAB (2013)
12. Lian, S., Liu, Z., Ren, Z.: Commutative encryption and watermarking in video compression. IEEE Trans. Circ. Syst. Video Technol. **17**(6), 774–778 (2007)
13. Liu, W., Li, H., Chen, Y., Zhu, H.: Lares: latency-reduced neighbour discovery for contagious diseases prevention. Int. J. Ad. Ubiq. Co. **16**(1), 3–13 (2014)
14. Luu, L., Teutsch, J., Kulkarni, R., Saxena, P.: Demystifying incentives in the consensus computer. In: CCS, pp. 706–719 (2015)
15. Mei, A., Stefa, J.: Give2get: forwarding in social mobile wireless networks of selfish individuals. IEEE Trans. Depend. Secure **9**(4), 569–582 (2012)
16. Mousa, H., Mokhtar, S.B., Hasan, O., Younes, O., Hadhoud, M., Brunie, L.: Trust management and reputation systems in mobile participatory sensing applications: a survey. Comput. Netw. **90**, 49–73 (2015)
17. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system, pp. 1–28 (2008). https://bitcoin.org/bitcoin.pdf
18. Ning, T., Yang, Z., Xie, X., Wu, H.: Incentive-aware data dissemination in delay-tolerant mobile networks. In: SECON, pp. 539–547. IEEE (2011)
19. Tarable, A., Nordio, A., Leonardi, E., Marsan, M.: The importance of being earnest in crowdsourcing systems. In: INFOCOM, pp. 2821–2829 (2015)
20. Wang, Y., Cai, Z., Yin, G., Gao, Y.: An incentive mechanism with privacy protection in mobile crowdsourcing systems. Comput. Netw. **102**, 157–171 (2016)
21. Zhong, S., Chen, J., Yang, Y.R.: Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: INFOCOM, pp. 1987–1997. IEEE (2003)
22. Zhu, H., Lin, X., Lu, R., Shen, X.S.: A secure incentive scheme for delay tolerant networks. In: ChinaCom, pp. 23–28. IEEE (2008)