A Collusion-Resistant Routing Scheme for Noncooperative Wireless Ad Hoc Networks

Sheng Zhong, *Associate Member, IEEE*, and Fan Wu, *Member, IEEE*

***Abstract*—In wireless ad hoc networks, routing needs cooperation of nodes. Since nodes often belong to different users, it is highly important to provide incentives for them to cooperate. However, most existing studies of the incentive-compatible routing problem focus on individual nodes' incentives, assuming that no subset of them would collude. Clearly, this assumption is not always valid. In this paper, we present a systematic study of collusion-resistant routing in noncooperative wireless ad hoc networks. In particular, we consider two standard solution concepts for collusion resistance in game theory, namely Group Strategyproofness and Strong Nash Equilibrium. We show that achieving Group Strategyproofness is impossible, while achieving Strong Nash Equilibrium is possible. More specifically, we design a scheme that is guaranteed to converge to a Strong Nash Equilibrium and prove that the total payment needed is bounded. In addition, we propose a cryptographic method that prevents profit transfer among colluding nodes, as long as they do not fully trust each other unconditionally. This method makes our scheme widely applicable in practice. Experiments show that our solution is collusion-resistant and has good performance.**

***Index Terms*—Collusion, routing, wireless ad hoc networks.**

I. Introduction

WIRELESS ad hoc networks have been widely used to achieve better connectivity at places where an infrastructure is not immediately available or cannot be directly used. The functioning of a wireless ad hoc network depends on the cooperation of the nodes in the network. For example, routing packets through the most cost-efficient path needs the information from each node about its cost for forwarding packets. In civilian ad hoc networks, nodes often belong to different individuals and have their own interests. Consequently, nodes may not always behave cooperatively unless incentives are provided.

The problem of incentive-compatible routing has received much attention [10], [32]–[34], [37]. Nevertheless, most existing solutions focus on the economic incentives of each *individual node*, assuming that no subset of nodes would collude. This assumption is not always valid in many practical scenarios.For example, consider an ad hoc network that uses a VCG-based payment scheme [2] to stimulate intermediate nodes to forward packets. Due to the property of VCG payment, an individual node cannot benefit from cheating in the routing protocol. However, when two or more nodes collaboratively cheat, they can benefit from cheating. Hence, the result may be that some colluding nodes get more utilities while the overall system performance degrades. Therefore, it is crucial to study how to achieve collusion resistance in incentive-compatible routing.

An elegant result on collusion resistance was obtained by Wang and Li in [33]. They showed that strategyproofness cannot be achieved when profit can be transferred between colluding nodes. While this result is elegant and crucial, there are fundamental questions about collusion resistance remaining unanswered. For example, in classic game theory, there are standard solution concepts for collusion resistance, like *Group Strategyproofness* and *Strong Nash Equilibrium*. Can these concepts be achieved in the routing of wireless ad hoc networks? The objective of this paper is to present a systematic study of collusion resistance to address such questions.

The major contributions of this paper are as follows:

- First, we show that the standard solution concept of Group Strategyproofness cannot be achieved in ad hoc networks. We prove this result *without assuming* that profit can be transferred between colluding nodes. This result indicates that we have to seek collusion resistance at a different level.
- Second, we show that the standard solution concept of Strong Nash Equilibrium can be achieved in ad hoc networks. In fact, we design a scheme in which all Nash equilibria are strong Nash equilibria. Therefore, regardless of which Nash equilibrium the system converges to, nodes cannot benefit from collusion.
- Third, we consider the total payment to all nodes in our scheme. We show that it has an upper bound that normally should not be much higher than the overall cost of lowest-cost path.
- Fourth, we study the prevention of profit transfer between colluding nodes, assuming they do not fully trust each other unconditionally.[1] In particular, we propose a method that makes it impossible for each node to convince other nodes about what action it has taken. Consequently, other nodes are not willing to transfer profit to this node in fear that this node may be cheating them.

Manuscript received October 03, 2007; revised April 29, 2008 and April 19, 2009; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Walrand. First published October 20, 2009; current version published April 16, 2010. This work was supported in part by NSF CNS-0524030 and CNS-0845149. Part of the results were presented at ACM MobiCom 2007.

S. Zhong is with the Computer Science and Engineering Department, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: szhong@buffalo.edu).

F. Wu was with the Computer Science and Engineering Department, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA. He is now with the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: fwu2@buffalo.edu).

Digital Object Identifier 10.1109/TNET.2009.2030325

[1]Note that the type of *collusion* we consider here is different from the type of *collusion* studied in cryptography, where all colluding parties are controlled by a single adversary and thus trust each other unconditionally. In the scenarios we consider, each colluding node is independent and actually has its own interest; the reason it colludes with other nodes is that it wants to maximize its own utility in this way. Therefore, in our scenarios, colluding nodes do not fully trust each other unconditionally.

- Finally, we evaluate our solution using extensive experiments. Simulations demonstrate that our scheme is resistant to collusion. Measurements of the overheads of our solution show that it has good efficiency.

The rest of the paper is organized as follows. Section II presents the technical preliminaries. The impossibility of achieving Group Strategyproofness is proved in Section III, and the scheme to achieve Strong Nash Equilibrium is given in Section IV. Most of Section V is dedicated to the study of preventing profit transfer between colluding nodes; the rest is used to cover the study of preventing collusions across communication sessions. Section VI gives our evaluation results. In Section VII, we discuss related work. We conclude the paper in Section VIII.

II. Technical Preliminaries

We use a graph $G = (V, E)$ to model a wireless ad hoc network, where V is the set of nodes, and $E \subseteq V \times V$ is the set of edges. We assume that G is biconnected.

Relaying data packet consumes nodes' battery power. Here, we ignore types of power consumption other than receiving and transmitting data packets, such as packet discarding, standby, and all operations on control packets, as they consume much less energy [11], [21]. For each node $v_i \in V$, there is a cost $c_i \in \mathbb{R}^+$ for relaying a unit of data to one of its neighbors. We allow power control in this paper. Each node's owner can choose the power level of his device when his device joins the network. In our model, therefore c_i can be different for different v_i. However, since our network is static (i.e., there is no mobile node), the cost c_i does not change from session to session. Note that the cost c_i can be either the deterministic cost under the binary link model, where a packet is always received if the transmission power is above a threshold, or the expected cost under the more realistic model, where a packet is received with a probability [8], [32], [35]. The cost c_i is a private information to the node v_i itself. It is also known as *type* in some previous papers.

We model the routing procedure as a strategic game, which we call the *routing game*. In a (unicast) *routing game*, suppose that the source node is S and the destination node is D. Then, the player set of the unicast routing game is $V - \{S, D\}$. We assume that there exists a secure network topology discovery protocol (*e.g.*, Link Layer Topology Discovery (LLTD) protocol) to enable nodes to know the topology of the network. In this game, when queried for cost, each player node v_i chooses an action, which is a claimed cost, based on its own cost: $a_i = \mathcal{A}_i(c_i)$. Note that a_i may not be equal to c_i, which is v_i's real cost. We also assume that the transmissions of claimed costs are secure, such that intermediate nodes cannot modify others' claimed costs. The claimed costs are delivered as control packets. In this paper, we do not consider the power consumed for transmissions of control packets, as mentioned earlier. Denote by a the profile of all players' actions (claimed cost): $a = (a_i)_{v_i \in V - \{S,D\}}$. This action profile decides a path for forwarding data from S to D. Each node v_i in this path receives a payment $p_i(a)$ from S for each unit of forwarded data. In addition, regardless of whether node v_i is in the path or not, it can also receive a one-time payment $p'_i(a)$ from S for the entire session. (Note that we do not study when and how payments $p_i(a)$ and $p'_i(a)$ should be collected in this paper. We adopt the assumption from [32] and [36] that there is a central authority who collects payments from the source node and guarantees secure distribution of payments to the forwarding nodes. We also adopt techniques from [37] for secure packet forwarding. This enables us to focus on the stage of routing.) The utility of node v_i is defined as the total payment node v_i receives minus its cost for forwarding data (if any). Formally, node v_i's utility is as follows:

$$u_i(a) = n \cdot \sigma_i(a) \cdot (p_i(a) - c_i) + p'_i(a).$$

In the above equation, $n \in \mathbb{N}^+$ is the number of units of data sent from S to D; $\sigma_i(a) = 1$ if node v_i is in the selected path for forwarding the data; $\sigma_i(a) = 0$ if node v_i is not. Clearly, the nodes $\{v_i | \sigma_i(a) = 1\}$ should form the path from S to D. Note that, in our game model, each session of communication is a game. Hence, we always consider multiple communications sessions as multiple games (except in Section V-B, where we extend the game model).

Before reviewing the solution concepts we use in this paper, we recall the definition of Strategyproof Equilibrium:

Definition 1: (Strategyproof Equilibrium [23], [31]) An action profile a^* is a *Strategyproof Equilibrium* if for all cost profile $c = (c_i)_{v_i \in V - \{S,D\}}$, for all action profile a, for all $n \in \mathbb{N}^+$

$$u_i(a_i^*, a_{-i}) \geq u_i(a_i, a_{-i}).$$

In the above, $a_i^* = c_i, \forall v_i \in V$.

Denote by a_C the profile of actions for a subset C of players: $a_C = (a_i)_{v_i \in C}$. Denote by $\overline{C}$ the complement set of C: $\overline{C} = V - \{S, D\} - C$. We have the standard solution concepts of Group Strategyproof Equilibrium and Strong Nash Equilibrium as follows.

Definition 2: (Group Strategyproof Equilibrium [16], [26]) An action profile a^* is a *Group Strategyproof Equilibrium* if for all nonempty subset C of player nodes, for all cost profile $c = (c_i)_{v_i \in V - \{S,D\}}$, for all action profile a, for all $n \in \mathbb{N}^+$, either for all $v_i \in C$

$$u_i(a_C^*, a_{\overline{C}}) = u_i(a_C; a_{\overline{C}})$$

or there exists a player node $v_i \in C$ such that

$$u_i(a_C^*, a_{\overline{C}}) > u_i(a_C; a_{\overline{C}}).$$

In the above, $a_i^* = c_i, \forall v_i \in V$.

Note that set of Group Strategyproof Equilibria is a subset of Strategyproof Equilibria.

Definition 3: (Strong Nash Equilibrium [23]) An action profile a^* is a *Strong Nash Equilibrium* if for all nonempty subset C of player nodes, for all cost profile $c = (c_i)_{v_i \in V - \{S,D\}}$, for all profile a_C of actions in the subset C, for all $n \in \mathbb{N}^+$, there exists a player node $v_i \in C$ such that

$$u_i(a_C^*, a_{\overline{C}}^*) \geq u_i(a_C; a_{\overline{C}}^*).$$

In reality, any practical solution to the routing game should satisfy additional requirements. For example, we should have *social efficiency*, which means that the total cost of the selected

path must be minimum. Also, each player node should have *individual rationality*, which means that its utility should be always greater than or equal to 0, since otherwise the player node would simply choose to remain out of the game. We combine these two requirements to define the admissibility of a solution.

Definition 4: (Admissibility) In a unicast routing game, suppose that a^* is a Group Strategyproof Equilibrium or a Strong Nash Equilibrium. We say a^* is *admissible* if the following two requirements are met for all cost profile $c = (c_i)_{v_i \in V - \{S,D\}}$:

- The nodes $\{v_i | \sigma_i(a^*) = 1\}$ form the lowest-cost path from S to D.
- For all $n \in \mathbb{N}^+$, for all player node v_i, $u_i(a^*) \geq 0$.

Using the above definitions, now we can formally state the main questions addressed in Sections III and IV. As the system designer, we have the freedom of choosing the functions $p_i()$, $p'_i()$, and $\sigma_i()$—the choice we make is called a *scheme*. Is there a way to design a scheme such that the system converges to an admissible Group Strategyproof Equilibrium or an admissible Strong Nash Equilibrium? (Here, convergence means the process in which all the nodes in the ad hoc network gradually change their behavior until they reach an equilibrium state.) Our answer is that the former is impossible, while the latter can be achieved.

III. Impossibility of Group Strategyproofness

In this section, we show that Group Strategyproofness cannot be achieved since it is contradictory to our requirement of admissibility.

Theorem 1: In any unicast routing game, if there is only one lowest-cost path from S to D, then there does not exist any admissible Group Strategyproof Equilibrium.

Proof: Clearly, any Group Strategyproof Equilibrium is also a Strategyproof Equilibrium. Therefore, for an arbitrary unicast routing game, we show that any Strategyproof Equilibrium a^* is not Group Strategyproof if it is admissible.

The basic idea of our proof is that we can carefully construct a colluding group of nodes and their actions, such that some colluding nodes can benefit from collusion without decreasing the other colluding nodes' utility. Logically, this suffices for the proof of nonexistence.

Denote by $\mathrm{LCP}(S, D, c)$ the lowest-cost path from node S to node D when the cost profile is c. We construct a player subset $C = \{i_0\} \cup \overline{\mathrm{LCP}(S, D, c)}$, where v_{i_0} is a node in $\mathrm{LCP(S, D, c)}$ and $\overline{\mathrm{LCP}(S, D, c)}$ is the set of nodes out of $\mathrm{LCP}(S, D, c)$. For all $v_i \in C$, we define

$$c'_i = c_i + p_{i_0}(a^*_C, a^*_{\overline{C}}) + p'_{i_0}(a^*_C, a^*_{\overline{C}}) + 1. \tag{1}$$

Before we prove our theorem, we first prove the following lemma.

Lemma 2: When c is the cost profile, for all $v_i \in C$

$$\sigma_i(a^*_C, a^*_{\overline{C}}) = \sigma_i(a_C, a^*_{\overline{C}})$$

where

$$a_C = (a^*_i(c'_i))_{v_i \in C}. \tag{2}$$

Proof: Since a^* is admissible, when c is the cost profile, clearly we have

$$\sigma_i(a^*_C, a^*_{\overline{C}}) = 1 \Leftrightarrow v_i \in \mathrm{LCP}(S, D, c).$$

On the other hand, considering a different scenario in which $(c'_C, c_{\overline{C}})$ is the cost profile, from (2) we can easily get that

$$\sigma_i(a_C, a^*_{\overline{C}}) = 1 \Leftrightarrow v_i \in \mathrm{LCP}(S, D, (c'_C, c_{\overline{C}})).$$

Furthermore, from (1) we know that

$$\mathrm{LCP}(S, D, c) = \mathrm{LCP}(S, D, (c'_C, c_{\overline{C}})).$$

Combining the above three equations, we have

$$\sigma_i(a^*_C, a^*_{\overline{C}}) = \sigma_i(a_C, a^*_{\overline{C}}).$$

■

Now, we come back to the proof of our theorem. Consider each $v_i \in C$, $i \neq i_0$. Clearly, $\sigma_i(a^*_C, a^*_{\overline{C}}) = 0$. By Lemma 2, this implies that

$$\sigma_i(a_C, a^*_{\overline{C}}) = 0. \tag{3}$$

Since a^* is Strategyproof, considering the scenario in which $(c'_C, c_{\overline{C}})$ is the cost profile, we get[2]

$$u_i(a^*_C, a^*_{\overline{C}}) \leq u_i(a_C, a^*_{\overline{C}}).$$

Since $\sigma_i(a^*_C, a^*_{\overline{C}}) = \sigma_i(a_C, a^*_{\overline{C}}) = 0$

$$p'_i(a^*_C, a^*_{\overline{C}}) \leq p'_i(a_C, a^*_{\overline{C}}). \tag{4}$$

When we put (3) and (4) together, we obtain that when c is the cost profile

$$u_i(a^*_C, a^*_{\overline{C}}) \leq u_i(a_C, a^*_{\overline{C}}). \tag{5}$$

Finally, we consider v_{i_0}. From (1), we have

$$np_{i_0}(a^*_C, a^*_{\overline{C}}) + p'_{i_0}(a^*_C, a^*_{\overline{C}}) < nc'_{i_0}. \tag{6}$$

From (2), since a^* is admissible, considering the scenario in which $(c'_C, c_{\overline{C}})$ is the cost profile, we have

$$nc'_{i_0} \leq np_{i_0}(a_C, a^*_{\overline{C}}) + p'_{i_0}(a_C, a^*_{\overline{C}}). \tag{7}$$

We put (6) and (7) together and obtain that

$$\begin{aligned} n(p_{i_0}(a^*_C, a^*_{\overline{C}}) - c_{i_0}) + p'_{i_0}(a^*_C, a^*_{\overline{C}}) \\ < n(p_{i_0}(a_C, a^*_{\overline{C}}) - c_{i_0}) + p'_{i_0}(a_C, a^*_{\overline{C}}). \end{aligned} \tag{8}$$

Since $\sigma_{i_0}(a^*_C, a^*_{\overline{C}}) = 1$, using (8) and Lemma 2, we get that when c is the cost profile

$$u_{i_0}(a^*_C, a^*_{\overline{C}}) < u_{i_0}(a_C, a^*_{\overline{C}}). \tag{9}$$

Equations (5) and (9) together imply that a^* is not Group Strategyproof. ■

[2]In fact, if we consider instead the scenario in which c is the cost profile, we can get the inequality in the other direction: $p'_i(a^*_C, a^*_{\overline{C}}) \geq p'_i(a_C, a^*_{\overline{C}})$. Hence, actually we have $p'_i(a^*_C, a^*_{\overline{C}}) = p'_i(a_C, a^*_{\overline{C}})$. However, to prove Theorem 1, it suffices to have (4).

IV. The Existence of a Strong Nash Equilibrium and How to Converge to One

In Section III, we have shown that in general we cannot guarantee the existence of admissible Group Strategyproof Equilibrium in the routing game, and thus clearly we cannot hope the system to converge to an admissible Group Strategyproof Equilibrium. Fortunately, we can design a scheme such that the system converges to an admissible Strong Nash Equilibrium. Note that the solution concept of Strong Nash Equilibrium is different from that of Group Strategyproof Equilibrium.

A. Scheme

The key idea of our design is discretization of costs. In practice, the cost c_i of each node has a finite precision. Therefore, without loss of generality, we assume that there is a very small real number $\epsilon \in \mathbb{R}^+$ such that for all player node v_i, c_i is a multiple of ϵ. Naturally, whenever a node claims its cost, we require that the claimed cost is also a multiple of ϵ. (Nevertheless, in our scheme, the payment to each node is not necessarily a multiple of ϵ—this is a very important feature of our scheme.) Based on this idea, we design a scheme in which each node makes a claim about its cost for forwarding a unit of data. If a node is in the lowest-cost path, our scheme gives it incentives to maximize its claimed cost (to the extent that it does not fall out of the lowest-cost path); if a node is out of the lowest-cost path, our scheme gives it incentives to minimize its claimed cost (to the extent that it does not fall into the lowest-cost path). Consequently, whenever the system converges to a Nash Equilibrium, each node in the lowest-cost path has a claimed cost equal to or slightly higher than its real cost, and each node out of the lowest-cost path has a claimed cost equal to or slightly lower than its real cost. Interestingly, we can show that such a Nash Equilibrium is actually a Strong Nash Equilibrium.

Specifically, in our scheme, the payment p_i for each unit of data is equal to the claimed cost of node v_i. Therefore, each node in the lowest-cost path has incentives to increase its claimed cost, as long as it remains in the lowest-cost path after the increase. In contrast, the one-time payment p_i' decreases in the claimed cost of node v_i. Therefore, each node out of the lowest-cost path has incentives to decrease its claimed cost, as long as it remains out of the lowest-cost path after the decrease. Of course, nodes in the lowest-cost path also receive one-time payments. We have to make sure that changes of one-time payments do not influence these nodes. To achieve this goal, we make all one-time payments smaller than ϵ. Hence, for all node v_i in the lowest-cost path, the total payment always increases whenever p_i increases (because the increase of p_i is at least ϵ and the decrease of p_i' is less than ϵ).

We emphasize that these are only some intuitive thoughts behind our design, which are not completely precise. For precise analysis, see the theorems, lemmas, and proofs we present.

Fig. 1 summarizes the details of our scheme. Given this detailed description of our scheme, now we can present the formal analysis of our scheme. We have three major results: 1) there exists a Nash equilibrium; 2) all Nash equilibria are admissible

Suppose S wants to some data to D.

- **[Initiation]** S initiates with broadcasting a query for forwarding cost with TTL t (t is a constant system parameter). On receiving the query, each node replies to S with a claimed cost $a_i = k \cdot \epsilon, k \in \mathbb{N}$; and rebroadcasts the query if the deducted $t(t = t - 1)$ is positive.
- **[Calculation]** S computes the lowest(-claimed)-cost path (LCP) to D using Dijkstra's algorithm [a]. If there is a tie, S breaks the tie according to the lexicographical order. For each node v_i in the LCP, S computes a one-time payment:

$$p_i'(a) = \frac{\epsilon}{1 + \max_{v_i \in LCP(S,D,(a_i', a_{\overline{\{i\}}}))} a_i'},$$

  Here, $\max_{v_i \in LCP(S,D,(a_i', a_{\overline{\{i\}}}))} a_i'$ is the largest cost v_i can claim, when each other node v_j still claims a_j, such that v_i remains in the LCP. For each node v_i not in the LCP, S computes another one-time payment:

$$p_i'(a) = \frac{\epsilon}{1 + a_i}.$$

- **[Transmission]** S starts data transmission and counts the number of packet sent as n.
- **[Compensation]** S ends the transmission and pays each node v_i with:

$$P_i = n \cdot \sigma_i(a) \cdot p_i(a) + p_i'(a),$$

  here $p_i(a) = a_i$.

[a]Note that it is also possible to do the calculation at D or any trusted node in the implementation. For convenience of presentation, we let S collect claimed costs and do the calculation in this paper.

Fig. 1. Scheme for achieving Strong Nash Equilibrium.

Strong Nash Equilibria; 3) for all Nash equilibria, there is an upper bound on the total payment to all nodes.

We show the existence of Nash equilibrium, when our scheme is used, by constructing a Nash equilibrium manually (Theorem 3). Since there may be a large number of Nash equilibria the system can converge to, it is crucial to prove that any achieved Nash equilibrium is admissible and strong. For the admissibility, we use Lemma 4 to show that any achieved Nash equilibrium is socially efficient, and we use Lemma 5 to show that the payments can cover the forwarding costs of intermediate nodes. Next, we prove by contradiction that any achieved Nash equilibrium is strong (Theorem 6). Finally, we study the upper bound of payment using a carefully designed alternative graph.

B. Existence of Nash Equilibrium

Theorem 3: If the scheme in Fig. 1 is used, then there exists a Nash equilibrium.

Proof: We construct an action profile a^* as follows. Initially, we set $a_i^* = c_i$ for each v_i. Then, for each player node $v_i \notin \mathrm{LCP}(S, D, c)$, if $a_i^* > 0$ and $\mathrm{LCP}(S, D, (a_i^* - \epsilon, a_{\overline{\{i\}}}^*)) = \mathrm{LCP}(S, D, a^*)$, we decrease a_i^* by ϵ; otherwise, keep the value of a_i^*. For each player node $v_i \in \mathrm{LCP}(S, D, c)$, if $\mathrm{LCP}(S, D, (a_i^* + \epsilon, a_{\overline{\{i\}}}^*)) = \mathrm{LCP}(S, D, a^*)$, we increase a_i^* by ϵ; otherwise, keep the value of a_i^*.

We repeat the above process until it does not make any change to any a_i^*. When the iteration stops, we get the action profile a^* we want.

We note that the above process will stop in a finite number of steps. Next, we show that a^* is a Nash equilibrium.

For each node v_i, we need to show that, for all a_i, $u_i(a_i^*, a^*_{\{i\}}) \geq u_i(a_i, a^*_{\{i\}})$. We distinguish two cases.

Case A: $v_i \notin \mathrm{LCP}(S,D,c) = \mathrm{LCP}(S,D,a^*)$. Then

$$u_i(a_i^*, a^*_{\{i\}}) = \frac{\epsilon}{1+a_i^*}.$$

If $a_i > a_i^*$, clearly $v_i \notin \mathrm{LCP}(S,D,(a_i, a^*_{\{i\}}))$ and

$$u_i(a_i, a^*_{\{i\}}) = \frac{\epsilon}{1+a_i} < \frac{\epsilon}{1+a_i^*} = u_i(a_i^*, a^*_{\{i\}}).$$

If $a_i < a_i^*$ (which is equivalent to $a_i \leq a_i^* - \epsilon$), by the above stopping criterion, we know that $v_i \in \mathrm{LCP}(S,D,(a_i, a^*_{\{i\}}))$. Thus

$$\begin{aligned} u_i(a_i,&a^*_{\{i\}}) \\ &= n \cdot \sigma_i(a_i, a^*_{\{i\}}) \cdot (p_i(a_i, a^*_{\{i\}}) - c_i) + p_i'(a_i, a^*_{\{i\}}) \\ &= n \cdot (a_i - c_i) + \frac{\epsilon}{1+\max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'} \\ &\leq n \cdot (a_i - c_i) + \epsilon \\ &\leq n \cdot (a_i^* - \epsilon - c_i) + \epsilon \\ &\leq -n\epsilon + \epsilon \\ &\leq 0 < u_i(a_i^*, a^*_{\{i\}}). \end{aligned}$$

Case B: $v_i \in \mathrm{LCP}(S,D,c) = \mathrm{LCP}(S,D,a^*)$. Then $u_i(a_i^*, a^*_{\{i\}}) = n(a_i^* - c_i) + \dfrac{\epsilon}{1+\max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'}$.

If $a_i < a_i^*$ (which is, again, equivalent to $a_i \leq a_i^* - \epsilon$), clearly $v_i \in \mathrm{LCP}(S, D(a_i, a^*_{\{i\}}))$ and

$$\begin{aligned} u_i(a_i,&a^*_{\{i\}}) \\ &= n \cdot \sigma_i(a_i, a^*_{\{i\}}) \cdot (p_i(a_i, a^*_{\{i\}}) - c_i) + p_i'(a_i, a^*_{\{i\}}) \\ &= n(a_i - c_i) + \frac{\epsilon}{1+\max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'} \\ &\leq n(a_i - c_i) + \epsilon \\ &\leq n(a_i^* - \epsilon - c_i) + \epsilon \\ &\leq n(a_i^* - c_i) \leq u_i(a_i^*, a^*_{\{i\}}). \end{aligned}$$

If $a_i > a_i^*$, by the stopping criterion, we know that $v_i \notin \mathrm{LCP}(S,D,(a_i, a^*_{\{i\}}))$. Thus

$$\begin{aligned} u_i(a_i, a^*_{\{i\}}) &= p_i'(a_i, a^*_{\{i\}}) \\ &= \frac{\epsilon}{1+a_i} \\ &< \frac{\epsilon}{1+\max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'} \\ &\leq u_i(a_i^*, a^*_{\{i\}}). \end{aligned}$$

In the above, the first inequality follows from the fact that $v_i \notin \mathrm{LCP}(S,D,(a_i, a^*_{\{i\}}))$ and thus

$$a_i > \max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'.$$

■

C. Admissible Strong Nash Equilibrium

Before we go to our proof that all Nash equilibria are admissible Strong Nash Equilibria, we need to establish two technical lemmas.

Lemma 4: If the above scheme is used, then for each Nash equilibrium a^*, $\mathrm{LCP}(S,D,c) = \mathrm{LCP}(S,D,a^*)$. That is, the lowest-cost path is always selected in all Nash equilibria.

Proof: We prove this lemma by contradiction. Suppose that there exists a Nash equilibrium a^* such that $\mathrm{LCP}(S,D,c) \neq \mathrm{LCP}(S,D,a^*)$. We distinguish two cases.

Case A: There exists v_i such that $v_i \in \mathrm{LCP}(S,D,c)$, $v_i \notin \mathrm{LCP}(S,D,a^*)$, $a_i^* > c_i$. Then, we consider v_i's utility when it claims the real cost c_i and all other nodes still remain with their equilibrium actions. If $v_i \in \mathrm{LCP}(S,D,(c_i, a^*_{\{i\}}))$

$$\begin{aligned} u_i(c_i, a^*_{\{i\}}) &= n(c_i - c_i) + \frac{\epsilon}{1+\max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'} \\ &> \frac{\epsilon}{1+a_i^*} = u_i(a_i^*, a^*_{\{i\}}). \end{aligned}$$

In the above, the inequality is due to fact that $v_i \notin \mathrm{LCP}(S,D,a^*)$ and thus $a_i^* > \max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\{i\}}))} a_i'$. This is contradictory to the fact that a^* is a Nash equilibrium. If $v_i \notin \mathrm{LCP}(S,D,(c_i, a^*_{\{i\}}))$, we have

$$u_i(c_i, a^*_{\{i\}}) = \frac{\epsilon}{1+c_i} > \frac{\epsilon}{1+a_i^*} = u_i(a_i^*, a^*_{\{i\}}).$$

Again, this is contradictory to the fact that a^* is a Nash equilibrium.

Case B: For all v_i such that $v_i \in \mathrm{LCP}(S,D,c)$, $v_i \notin \mathrm{LCP}(S,D,a^*)$, we have $a_i^* \leq c_i$. Assume that when $\mathrm{LCP}(S,D,c)$ and $\mathrm{LCP}(S,D,a^*)$ have the same claimed cost, the tie-breaking rule chooses $\mathrm{LCP}(S,D,a^*)$ over $\mathrm{LCP}(S,D,c)$. (If the tie-breaking rule chooses $\mathrm{LCP}(S,D,c)$ over $\mathrm{LCP}(S,D,a^*)$, we have a similar proof, which we skip to save space.) Then, we know that

$$\begin{aligned} \sum_{v_i \in \mathrm{LCP}(S,D,a^*)} a_i^* &\leq \sum_{v_i \in \mathrm{LCP}(S,D,c)} a_i^* \\ &= \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \notin \mathrm{LCP}(S,D,a^*)} a_i^* \\ &\quad + \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \in \mathrm{LCP}(S,D,a^*)} a_i^* \\ &\leq \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \notin \mathrm{LCP}(S,D,a^*)} c_i \\ &\quad + \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \in \mathrm{LCP}(S,D,a^*)} a_i^* \\ &\leq \sum_{v_i \notin \mathrm{LCP}(S,D,c) \wedge v_i \in \mathrm{LCP}(S,D,a^*)} c_i \\ &\quad + \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \in \mathrm{LCP}(S,D,a^*)} a_i^*. \end{aligned}$$

Using the above inequality, we can show that there exists v_i such that $v_i \notin \mathrm{LCP}(S,D,c)$, $v_i \in \mathrm{LCP}(S,D,a^*)$, $a_i^* < c_i$ (see

below). Therefore

$$\begin{aligned} u_i(a_i^*, a^*_{\overline{\{i\}}}) &= n(a_i^* - c_i) + \frac{\epsilon}{1 + \max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\overline{\{i\}}}))} a_i'} \\ &< n(a_i^* - c_i) + \epsilon \\ &\le -n\epsilon + \epsilon \\ &\le 0 \\ &\le u_i(c_i, a^*_{\overline{\{i\}}}) \end{aligned}$$

which is contradictory to that a^* is a Nash equilibrium.

Finally, we give a proof that there exists v_i such that $v_i \notin \mathrm{LCP}(S,D,c)$, $v_i \in \mathrm{LCP}(S,D,a^*)$, $a_i^* < c_i$. Suppose that this is not true. Then, using (10), we get that, for all v_i such that $v_i \notin \mathrm{LCP}(S,D,c)$ and $v_i \in \mathrm{LCP}(S,D,a^*)$, $a_i^* = c_i$. Since

$$\begin{aligned} &\sum_{v_i \notin \mathrm{LCP}(S,D,c) \wedge v_i \in \mathrm{LCP}(S,D,a^*)} a_i^* \\ &\quad \le \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \notin \mathrm{LCP}(S,D,a^*)} a_i^* \end{aligned} \tag{10}$$

we get that

$$\begin{aligned} &\sum_{v_i \notin \mathrm{LCP}(S,D,c) \wedge v_i \in \mathrm{LCP}(S,D,a^*)} c_i \\ &\quad \le \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \notin \mathrm{LCP}(S,D,a^*)} a_i^* \\ &\quad \le \sum_{v_i \in \mathrm{LCP}(S,D,c) \wedge v_i \notin LCP(S,D,a^*)} c_i \end{aligned}$$

which means that the real cost of $\mathrm{LCP}(S,D,a^*)$ is not more than that of $\mathrm{LCP}(S,D,c)$. This is impossible because even when their costs are equal, the tie-breaking rule should not choose $\mathrm{LCP}(S,D,c)$ as the lowest-cost path. ■

Lemma 5: If the abovementioned scheme is used, then for all Nash equilibrium a^*, we have that $a_i^* \ge c_i \Leftrightarrow v_i \in \mathrm{LCP}(S,D,c)$ and that $a_i^* \le c_i \Leftrightarrow v_i \notin \mathrm{LCP}(S,D,c)$.

Proof: We only need to show that $v_i \in \mathrm{LCP}(S,D,c) \Rightarrow a_i^* \ge c_i$ and that $v_i \notin \mathrm{LCP}(S,D,c) \Rightarrow a_i^* \le c_i$, which are equivalent to this lemma.

First, we prove $v_i \in \mathrm{LCP}(S,D,c) \Rightarrow a_i^* \ge c_i$ by contradiction. Suppose that there exists $v_i \in \mathrm{LCP}(S,D,c)$, such that $a_i^* < c_i$. Since $\mathrm{LCP}(S,D,c) = \mathrm{LCP}(S,D,a^*)$ (by Lemma 4), v_i's equilibrium utility is

$$\begin{aligned} u_i(a_i^*, a^*_{\overline{\{i\}}}) &= n(a_i^* - c_i) + \frac{\epsilon}{1 + \max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\overline{\{i\}}}))} a_i'} \\ &\le -n\epsilon + \frac{\epsilon}{1 + \max_{v_i \in \mathrm{LCP}(S,D,(a_i', a^*_{\overline{\{i\}}}))} a_i'} \\ &< -n\epsilon + \epsilon \\ &\le 0 \end{aligned}$$

which indicates that v_i can increase its utility by declaring a cost that brings itself out of the LCP. This is contradictory to the fact that a^* is a Nash equilibrium.

Next, we prove $v_i \notin \mathrm{LCP}(S,D,c) \Rightarrow a_i^* \le c_i$, also by contradiction. Suppose that there exists $v_i \notin \mathrm{LCP}(S,D,c)$, such that $a_i^* > c_i$. Since $\mathrm{LCP}(S,D,c) = \mathrm{LCP}(S,D,a^*)$ (by Lemma 4), v_i has an equilibrium utility

$$u_i(a_i^*, a^*_{\overline{\{i\}}}) = \frac{\epsilon}{1 + a_i^*}.$$

We claim that v_i can always increase its utility by declaring its real cost c_i: If $v_i \in \mathrm{LCP}(S,D,(c_i, a^*_{\overline{\{i\}}}))$, then

$$\begin{aligned} u_i(c_i, a^*_{\overline{\{i\}}})) &= n(c_i - c_i) + \frac{\epsilon}{1 + \max_{v_i \in LCP(S,D,(a_i', a^*_{\overline{\{i\}}}))} a_i'} \\ &> \frac{\epsilon}{1 + a_i^*} \\ &= u_i(a_i^*, a^*_{\overline{\{i\}}}). \end{aligned}$$

If $v_i \notin \mathrm{LCP}(S,D,(c_i, a^*_{\overline{\{i\}}}))$, then

$$u_i(c_i, a^*_{\overline{\{i\}}})) = \frac{\epsilon}{1 + c_i} > \frac{\epsilon}{1 + a_i^*} = u_i(a_i^*, a^*_{\overline{\{i\}}}).$$

This completes the proof. ■

Now, we are ready to show that all Nash equilibria are admissible Strong Nash Equilibria.

Theorem 6: If the above scheme is used, then all Nash equilibria are admissible Strong Nash Equilibria.

Proof: (Sketch) It is clear from Lemmas 4 and 5 that all Nash equilibria are admissible. Then, we only need to prove that all Nash equilibria are strong. We prove it by contradiction.

Suppose that there exists a Nash equilibrium a^* that is not strong. Then, there exists $C \subseteq V$ and an action profile a_C of C such that every node in C can increase its utility when they use a_C.

First, we show by contradiction that for all node v_i, if $v_i \in \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}}))$, then $v_i \in \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}}))$. Suppose that there exists v_i, $v_i \in \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}}))$, $v_i \notin \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}}))$. Assume that the tie-breaking rule prefers $\mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}}))$ to $\mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}}))$ when their claimed costs are equal. (We have a similar proof when the tie-breaking rule prefers $\mathrm{LCP}(S,D,(a_i, a^*_{\overline{C}}))$.) Then, we have

$$\begin{aligned} &\sum_{v_i \in \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}})), v_i \in C} a_i + \sum_{v_i \in \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}})), v_i \notin C} a_i^* \\ &\quad < \sum_{v_i \in \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}})), v_i \in C} a_i \\ &\qquad + \sum_{v_i \in \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}})), v_i \notin C} a_i^* \\ &\quad \Rightarrow \sum_{v_i \in \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}})), v_i \notin \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}})), v_i \in C} a_i \\ &\qquad + \sum_{v_i \in \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}})), v_i \notin \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}})), v_i \notin C} a_i^* \\ &\quad < \sum_{v_i \in \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}})), v_i \notin \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}})), v_i \in C} a_i \\ &\qquad + \sum_{v_i \in \mathrm{LCP}(S,D,(a_C^*, a^*_{\overline{C}})), v_i \notin \mathrm{LCP}(S,D,(a_C, a^*_{\overline{C}})), v_i \notin C} a_i^*. \end{aligned}$$

Since

$$\sum_{v_i\in \mathrm{LCP}(S,D,(a_C,a^*_{\overline{C}})),v_i\notin \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}}))} a_i^* \geq \sum_{v_i\in \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})),v_i\notin \mathrm{LCP}(S,D,(a_C,a^*_{\overline{C}}))} a_i^*$$

we have

$$\sum_{v_i\in \mathrm{LCP}(S,D,(a_C,a^*_{\overline{C}})),v_i\notin \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})),v_i\in C} (a_i-a_i^*) < \sum_{v_i\in \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})),v_i\notin \mathrm{LCP}(S,D,(a_C,a^*_{\overline{C}})),v_i\in C} (a_i-a_i^*). \tag{11}$$

We can easily show that, for all $v_i \in \mathrm{LCP}(S,D,(a_C,a^*_{\overline{C}}))$, $v_i \notin \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}}))$, $v_i \in C$, $a_i - a_i^* \geq 0$: Otherwise, $a_i - a_i^* < 0$, which implies that

$$\begin{aligned} u_i(a_C,a^*_{\overline{C}}) &= n(a_i-c_i)+\frac{\epsilon}{1+\max_{v_i\in \mathrm{LCP}(S,D,(a'_i,a_{C-\{i\}},a^*_{\overline{C}}))} a'_i} \\ &< n(a_i-c_i)+\epsilon \\ &\leq n(a_i^*-\epsilon-c_i)+\epsilon \\ &\leq -n\epsilon+\epsilon \\ &\leq 0 \\ &\leq u_i(a^*_C,a^*_{\overline{C}}). \end{aligned}$$

This is contradictory to our assumption. Similarly, we can easily show that, for all $v_i \notin \mathrm{LCP}(S,D,(a_C,a^*_{\overline{C}}))$, $v_i \in \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}}))$, $v_i \in C$, $a_i - a_i^* \leq 0$.

Combining the above two results with (11), we get a contradiction. Therefore, we must have $v_i \in \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})) \Rightarrow v_i \in \mathrm{LCP}(S,D,(a_C,a^*_C))$. This actually means

$$\mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})) = \mathrm{LCP}(S,D,(a_C,a^*_C)). \tag{12}$$

Using (12), from $\forall v_i \in C$, $u_i(a^*_C,a^*_{\overline{C}}) < u_i(a_C,a^*_{\overline{C}})$ we can easily get that

$$\begin{aligned} v_i \in C \wedge v_i \in \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})) &\Leftrightarrow a_i^* < a_i \\ v_i \in C \wedge v_i \notin \mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})) &\Leftrightarrow a_i^* > a_i. \end{aligned}$$

From the above result, it is not hard to get that $\mathrm{LCP}(S,D,(a^*_C,a^*_{\overline{C}})) = \mathrm{LCP}(S,D,(a_i^*,a^*_{\overline{\{i\}}}))$. Therefore, if $v_i \in \mathrm{LCP}(S,D,(a_i^*,a^*_{\overline{\{i\}}}))$

$$\begin{aligned} u_i(a_i^*,a^*_{\overline{\{i\}}}) &= n(a_i^*-c_i)+\frac{\epsilon}{1+\max_{v_i\in \mathrm{LCP}(S,D,(a'_i,a^*_{\overline{\{i\}}}))} a'_i} \\ &< n(a_i-c_i)+\frac{\epsilon}{1+\max_{v_i\in \mathrm{LCP}(S,D,(a'_i,a^*_{\overline{\{i\}}}))} a'_i} \\ &= u_i(a_i,a^*_{\overline{\{i\}}}), \end{aligned}$$

which is contradictory to that a^* is a Nash equilibrium. If $v_i \notin \mathrm{LCP}(S,D,(a_i^*,a^*_{\overline{\{i\}}}))$

$$u_i(a_i^*,a^*_{\overline{\{i\}}}) = \frac{\epsilon}{1+a_i^*} <, \frac{\epsilon}{1+a_i} = u_i(a_i,a^*_{\overline{\{i\}}})$$

which is also contradictory to that a^* is a Nash equilibrium. ■

D. Upper Bound on Payment

So far, we have shown that any Nash equilibrium is resistant to collusion when our scheme is used. A natural question is how much is needed to pay the nodes in these equilibria. Now, we show that the total payment needed in any Nash equilibrium actually has an upper bound, which is based on the *alternative graph* we define below.

Definition 5: In the network G, for source node S and destination D, we define the (S,D)-*alternative graph* as a *directed* graph $G' = (V,E')$, where

$$\begin{aligned} E' = \{&(v_i,v_j)|(v_i,v_j)\in E; \\ &v_i \notin \mathrm{LCP}(S,D,c) \text{ or } v_j \notin \mathrm{LCP}(S,D,c)\} \\ \cup\{&(v_i,v_j)|(v_i,v_j)\in \mathrm{LCP}(S,D,c) \\ &\text{and } v_j \text{ is closer to } S \text{ than } v_i \text{ in } G\}. \end{aligned}$$

Note that an alternative graph is always a subgraph of G (if we view G as a directed graph such that $(v_i,v_j)\in E \Leftrightarrow (v_j,v_i)\in E$), so every path in the alternative graph is also a path in G. Furthermore, since the alternative graph has kept the vast majority of edges in G, we can expect that a lot of paths in G still remain in the alternative graph.

Definition 6: An (S,D)-*alternative path* is a path from S to D in the (S,D)-alternative graph such that for any node v_i in this path, if $v_i \in \mathrm{LCP}(S,D,c)$, then either the node preceding v_i in this path is also in $\mathrm{LCP}(S,D,c)$, or the node following v_i in this path is also in $\mathrm{LCP}(S,D,c)$.

The following theorem guarantees that alternative paths exist.

Theorem 7: For every pair of (S,D), there exists an (S,D)-alternative path.

Proof: We construct an (S,D)-alternative path as follows. Without loss of generality, suppose $\mathrm{LCP}(S,D,c) = Sv_1v_2\ldots v_{\mathcal{L}}D$, where nodes are sorted according to their order in the path from S to D. Since G is biconnected, there must be a path P_1 from S to D that does not go through v_1. Let v_{i_1} be the first node in P_1 that is in $\mathrm{LCP}(S,D,c)$, and P'_1 be the part of P_1 from S to v_{i_1} (including v_{i_1}). Then, we can construct $P'_2,P'_3,\ldots,P'_J$ in the following way: Suppose we already have P'_1 through P'_j, where P'_j ends at node v_{i_j}. Since G is biconnected, there is a path P_{j+1} from v_{i_j-1} to D that does not go through v_{i_j}. Let $v_{i_{j+1}}$ be the first node on P_{j+1} that is in $\{v_\ell|\ell > i_j\}$. Let $v_{i'_{j+1}}$ be the last node before $v_{i_{j+1}}$ on P_{j+1} that is in $\{v_\ell|\ell < i_j\}$. Therefore, P'_{j+1} starts from v_{i_j}, going toward $v'_{i_{j+1}}$ along $\mathrm{LCP}(S,D,c)$, and then follows P_{j+1} to reach $v_{i_{j+1}}$. The last one of this sequence, P'_J, must end at D. Therefore, $P'_1P'_2\ldots P'_J$ is the (S,D)-alternative path we have constructed. It is not hard to verify that it is indeed an (S,D)-alternative path. ■

Definition 7: The *lowest-cost alternative path* for (S, D), denoted by $\mathrm{LCAP}(S, D, c)$, is the (S, D)-alternative path with the lowest cost.

Normally, the lowest-cost alternative path should not have a much higher cost than the lowest-cost path in G. Below, we show that it is an upper bound on the total payment.

Theorem 8: If our scheme is used, in any Nash equilibrium a^*, the total payment is

$$\sum_{v_i \in V} (p_i(a^*) + p'_i(a^*)) \leq \sum_{v_i \in \mathrm{LCAP}(S,D,c)} c_i + |V| \cdot \epsilon.$$

Proof: Again, without loss of generality, suppose $\mathrm{LCP}(S, D, c) = Sv_1v_2 \ldots v_{\mathcal{L}}D$. By the definition of (S, D)-alternative path, we can write $\mathrm{LCAP}(S, D, c)$ as

$$\begin{array}{lll} S & P_1 & v_{i_1} v_{i_1 - 1} \ldots v_{i'_1} \\ & P_2 & v_{i_2} v_{i_2 - 1} \ldots v_{i'_2} \\ & \ldots & \ldots \\ & P_J & D \end{array}$$

where all nodes in all P_j are out of $\mathrm{LCP}(S, D, c)$, and for every j, $i_j > i'_j$. Hence, (letting $v_{i'_0} = S$ and $v_{i_J} = D$)

$$\begin{aligned} \sum_{v_i \in \mathrm{LCAP}(S,D,c)} c_i &= \sum_{j=1}^{J} \sum_{v_i \in P_j} c_i + \sum_{j=1}^{J-1} \sum_{i=i'_j}^{i_j} c_i \\ &\geq \sum_{j=1}^{J} \sum_{v_i \in P_j} a_i^* \geq \sum_{j=1}^{J} \sum_{i=i'_{j-1}+1}^{i_j - 1} a_i^* \\ &\geq \sum_{j=1}^{J} \sum_{i=i_{j-1}}^{i_j - 1} a_i^* = \sum_{i=1}^{\mathcal{L}} a_i^* = \sum_{v_i \in \mathrm{LCP}(S,D,c)} p_i(a_i^*). \end{aligned}$$

Consequently, it is easy to see

$$\begin{aligned} &\sum_{v_i \in V} (p_i(a^*) + p'_i(a^*)) \\ &\quad = \sum_{v_i \in \mathrm{LCP}(S,D,a^*)} p_i(a^*) + \sum_{v_i \notin \mathrm{LCP}(S,D,a^*)} p_i(a^*) \\ &\qquad + \sum_{v_i \in V} p'_i(a^*)) \\ &\quad \leq \sum_{v_i \in \mathrm{LCAP}(S,D,c)} c_i + |V| \cdot \epsilon. \end{aligned}$$

■

V. Preventing Profit Transfer and Multisession Collusion

As we have mentioned, the standard solution concepts of Group Strategyproofness and Strong Nash Equilibrium are applicable if profit cannot be transferred between colluding nodes. In many practical scenarios, the assumption of no profit transfer is not immediately valid. To make our results applicable in those scenarios, we propose a method to prevent colluding nodes from transferring profit to each other, as long as they do not fully trust each other unconditionally. (Note that in civilian applications, nodes typically do not trust each other unconditionally, unless they belong to the same user.)

The main idea of our method is that we can make it impossible for colluding nodes to convince each other that they have taken the actions required by the collusion. For example, imagine that nodes v_1 and v_2 are trying to collude. They have a deal. If v_1 takes action a_1 and v_2 takes action a_2, then v_1 will transfer a profit of 7 to v_2. Suppose both of them follow the requirement of the deal. Then v_1 has an increase of 10 in utility, but v_2 has a decrease of 5 in utility. Therefore, v_1 would like to transfer a profit of 7 to v_2, such that both of them benefit from the collusion. However, the possibility of this profit transfer depends on if v_2 can convince v_1 about its action, so we design a method to make it impossible for v_2 to convince v_1 that it indeed takes action a_2. When v_2 claims that it has taken the action a_2, actually it might have taken another action a'_2. In this case, v_1's utility has only increased by 1 and v_2's utility has only decreased by 2. If v_1 trusts v_2's claim (of having taken action a_2) and transfers 7 to v_2, then v_1 actually loses 6 in utility while v_2 gains 5. If our method is used, then v_1 has no way to trust v_2's claim and becomes unwilling to transfer profit to v_2. In this way, all colluding nodes become unwilling to transfer profit, and the assumption of no profit transfer becomes valid.[3]

To implement our idea and develop our method, we need to consider how a node can convince other nodes about its own action. There are two basic approaches: Either the node convinces other nodes by showing messages it has sent, or the node does so by showing messages it has received. (Of course, it can also use a combination of the two basic approaches.) Among the sent messages, the only one related to its own action is its message to the source node S, which contains its claimed cost. The node may attempt to convince other nodes about its action by showing this message, but we can easily defeat its attempt as follows: We allow each node to update its claimed cost by sending an additional message to the sender. Therefore, even if other nodes see a (digitally signed) message with claimed cost, they still do not know what is the claimed cost recognized by the source node S because they have no idea whether this node has updated its claimed cost or not.

However, the other approach is harder to prevent. In particular, there is a message received by the node that contains information about its own action—the payment message from the source node S. Since the amount of payment is decided by the claimed cost, showing this payment message to other nodes can indirectly prove the node's claimed cost that is recognized by the sender. To deal with this difficulty, we propose a new cryptographic technique called *restricted verifier signature*.[4]

[3]One may suggest that v_1 should transfer 7 to v_2 only after the path and payments outcome is what they expected. However, in this case, v_1 can easily cheat v_2, for example, by taking action a'_1 such that, with (a'_1, a_2), v_1 gets an increase of 9 in its utility but v_2 gets a decrease of 4 in its utility. Since (a'_1, a_2) decides a different path and different payments outcome, v_1 can decline to transfer anything to v_2 when v_2 takes action a_2.

[4]Restricted verifier signature is closely related to the well known *designated verifier signature* and *multiple designated verifier signature* [18], but is different from both of them. Designated verifier signature schemes allow only one participant to verify the signature. Multiple designated verifier signature schemes allow more than one participants to verify the signature, but they require that each such participant should be able to simulate the signature, which is not the case with restricted verifier signature.

When the source node S makes a payment to a player node, it signs its payment using a restricted verifier signature scheme. Unlike traditional digital signatures, this restricted verifier signature can be verified *only by the player node (i.e., the payee) and a central bank*.[5] The player node can verify the signature to see that the payment is valid. When the node brings this payment to the bank, the bank can also verify the signature before clearing the transaction. Nevertheless, the restricted verifier signature scheme guarantees that the player node cannot use this signed payment to convince other nodes about its own action, because other nodes have no way to verify the signature—they would suspect that this node might have forged the signature to cheat them. Below, we outline the requirements for a restricted verifier signature scheme. We do not give a concrete scheme in this paper, but we conjecture that it is not hard to modify some existing signature scheme, especially some existing designated verifier signature scheme, to obtain a restricted verifier signature scheme. After the discussions of designated verifier signature, we briefly address the problem of preventing collusion across multiple sessions.

A. Restricted Verifier Signature Scheme

A restricted verifier signature scheme consists of three spaces and five algorithms: a key space KYES, a message space MSGS, a signature space SIGS; a key generation algorithm KeyGen, a signing algorithm Sign, a node's verification algorithm $\mathsf{NVerify}$, a bank's verification $\mathsf{BVerify}$, a node's simulation algorithm Sim. (Here, $\mathsf{NVerify}$ and $\mathsf{BVerify}$ may be the same algorithm, but for generality we allow them to be different.)

Intuitively, a restricted verifier signature scheme works as follows. First, KeyGen is executed, with a security parameter (i.e., the length of a key) as input, and it outputs the key pair (x_i, y_i) for each node v_i, where x_i is the private key and y_i is the public key; in addition, KeyGen outputs (x_B, y_B), where x_B is the bank's private key and y_B is the bank's public key. When S makes a payment to node v_j, S uses Sign and x_S to compute a signature on the payment. Upon receiving the payment, node v_j uses $\mathsf{NVerify}$, y_S and x_j to verify the signature, and then forwards the payment to the bank. The bank uses $\mathsf{BVerify}$, y_S, and x_B to verify the signature. Note that the simulation algorithm Sim is not directly used in the above—it is needed purely for security purpose, as we describe below.

There are four requirements for a restricted verifier signature scheme:

1) It must be *correct* in the sense that a valid signature can always be verified by the node v_j and the central bank.
2) Any signature accepted by v_j must contain a valid payment that will be honored by the bank.
3) The signature cannot be forged.
4) Any party other than v_j and the bank cannot verify the signature.

If a restricted verifier signature scheme satisfies all these four requirements, then it can help us prevent profit transfers.

Now, we formally define these four requirements.

[5]Note that using virtual currency requires the existence of a central bank. Our method does *not* require the bank to be online when a payment is made, although the bank is needed when the payment is finally cleared.

Definition 8: A restricted verifier signature scheme $(\mathsf{KYES}, \mathsf{MSGS}, \mathsf{SIGS}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{NVerify}, \mathsf{BVerify}, \mathsf{Sim})$ is correct if for

$$(\ldots, (x_S, y_S), \ldots, (x_j, y_j), \ldots, (x_B, y_B), \ldots) \leftarrow \mathsf{KeyGen}(\mathsf{KeyLen})$$

for all message $m \in \mathsf{MSGS}$

$$\mathsf{NVerfiy}_{y_S, x_j}(m, \mathsf{Sign}_{x_S}(m)) = \mathsf{accept}$$

and

$$\mathsf{BVerfiy}_{y_S, x_B}(m, \mathsf{Sign}_{x_S}(m)) = \mathsf{accept}.$$

Definition 9: A restricted verifier signature scheme $(\mathsf{KYES}, \mathsf{MSGS}, \mathsf{SIGS}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{NVerify}, \mathsf{BVerify}, \mathsf{Sim})$ satisfies the binding property if for

$$(\ldots, (x_S, y_S), \ldots, (x_j, y_j), \ldots, (x_B, y_B), \ldots) \leftarrow \mathsf{KeyGen}(\mathsf{KeyLen})$$

for all message $m \in \mathsf{MSGS}$, for all $\sigma \in \mathsf{SIGS}$ such that

$$\mathsf{NVerfiy}_{y_S, x_j}(m, \sigma) = \mathsf{accept}$$

we have that

$$\mathsf{BVerfiy}_{y_S, x_B}(m, \sigma) = \mathsf{accept}.$$

Definition 10: A restricted verifier signature scheme $(\mathsf{KYES}, \mathsf{MSGS}, \mathsf{SIGS}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{NVerify}, \mathsf{BVerify}, \mathsf{Sim})$ is existentially unforgeable if for

$$(\ldots, (x_S, y_S), \ldots, (x_j, y_j), \ldots, (x_B, y_B), \ldots) \leftarrow \mathsf{KeyGen}(\mathsf{KeyLen})$$

for all probabilistic polynomial-time algorithm Adv, for all polynomial $\mathsf{poly}()$, for all sufficiently large KeyLen

$$\Pr[\mathsf{NVerfiy}^{\mathsf{SO}}_{y_S, x_j}(\mathsf{Adv}(\ldots, y_S, \ldots, y_j, \ldots, y_B, \ldots)) = \mathsf{accept}] < \frac{1}{\mathsf{poly}(\mathsf{KeyLen})}$$

and

$$\Pr[\mathsf{BVerfiy}^{\mathsf{SO}}_{y_S, x_B}(\mathsf{Adv}(\ldots, y_S, \ldots, y_j, \ldots, y_B, \ldots)) = \mathsf{accept}] < \frac{1}{\mathsf{poly}(\mathsf{KeyLen})}$$

where SO is a signing oracle that replies querie messages with the corresponding signatures of S.

In the above definition, it is required that the output of Adv in an execution cannot be a query of Adv to SO in this execution together with the corresponding reply.

Definition 11: A restricted verifier signature scheme $(\mathsf{KYES}, \mathsf{MSGS}, \mathsf{SIGS}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{NVerify}, \mathsf{BVerify}, \mathsf{Sim})$ is secure against unauthorized verifiers if for

$$(\ldots, (x_S, y_S), \ldots, (x_j, y_j), \ldots, (x_B, y_B), \ldots) \leftarrow \mathsf{KeyGen}(\mathsf{KeyLen})$$

for all message $m \in \mathsf{MSGS}$

$$\sigma \leftarrow \mathsf{Sign}_{x_S}(m) \quad \text{and} \quad \sigma' \leftarrow \mathsf{Sim}_{x_j}(m)$$

are computationally indistinguishable.[6]

B. Prevention of Collusion Across Multiple Sessions

So far, we have focused on a single session of communications. Now, we formally show that we can prevent collusions across multiple sessions. It is important to note that our proof is in a repeated game model, which is an extension of the strategic game model in Section II. The reason for extending the model is to accommodate the consideration of collusions across two or more sessions because the strategic game model in Section II applies to only a single session.

To be precise, in the extended model, we have an infinitely repeated game with discounting, where each stage is a strategic game defined in Section II. We slightly modify the symbols used by adding a superscript (τ) to each variable in stage t. For example, the utility of player node v_i in stage τ is denoted by $u_i^{(\tau)}$. The total utility of v_i is

$$u_i^{\mathbf{total}} = \sum_{\tau=1}^{\infty} \delta^{\tau-1} u_i^{(\tau)}$$

where $\delta < 1$ is a constant called the discounting factor.

For repeated games, we often need to consider *histories*. Here, a history is defined as one or more continuous stages starting from the beginning of the entire game; in these stages, all players' actions have been chosen and fixed. The number of stages in a history is usually called its length.

In a repeated game, a *strategy* of a player specifies what action the player should choose after each possible history. For example, when player v_i uses strategy s_i, we can write v_i's action after history H as $s_i(H)$.

Clearly, total utilities of all players are decided by the strategy profile of all players. Denote by s ($s = (s_1, \ldots, s_n)$) the profile of all players' strategies. To emphasize this, we can write the total utility of v_i as $u_i^{\mathbf{total}}(s)$. Similarly, the utilities of all players in a stage are decided by all players' actions chosen by their strategies. For example, for a history H of length L, the utility of v_i in the stage after history H can be written as $u_i^{(L+1)}(s_1(H), \ldots, s_n(H))$.

In this extended model of repeated games, we have the following theorem.

Theorem 9: In the extended model, assume our scheme is used. Let $s^\star$ be the strategy profile in which all players follow the protocol in all stages. For an arbitrary colluding player set V_C ($|V_C| \geq 2$) using strategy profile $(s_i)_{v_i \in V_C}$, such that

$$\sum_{v_i \in V_C} u_i^{\mathsf{total}}(s) > \sum_{v_i \in V_C} u_i^{\mathsf{total}}(s^\star)$$

[6]Here, being *computationally indistinguishable* means that σ cannot be distinguished from σ' by any polynomial-time adversary. See [15] for the precise definition.

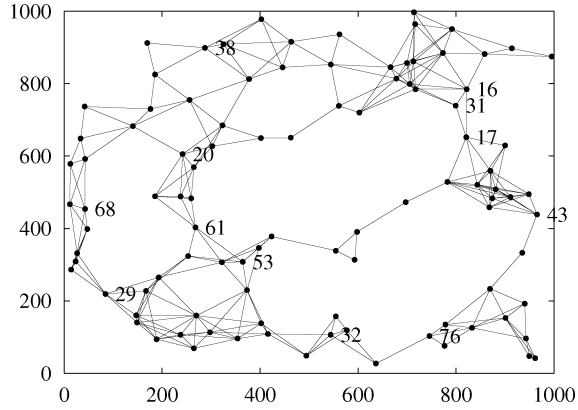


Fig. 2. Topology of the random generated network.

there exist $v_j, v_\ell \in V_C$ and $s'_j \neq s_j$ such that

$$u_j^{\mathsf{total}}(s') > u_j^{\mathsf{total}}(s), \quad \text{and} \quad u_\ell^{\mathsf{total}}(s') < u_\ell^{\mathsf{total}}(s)$$

where s is the strategy profile of all players obtained by replacing the colluding players' strategies in $s^\star$ with $(s_i)_{v_i \in V_C}$, and s' is the strategy profile of all players obtained by replacing v_j's strategy in s with s'_j.

(To save space, we skip the proof, which is not hard.)

Intuitively, Theorem 9 considers a group of player nodes that try to collude across multiple sessions. We restrict our attention to the case in which their total utility is increased because, otherwise, clearly there is no incentive for the nodes to collude. In this case, we find that at least one of the nodes (v_j in the theorem) has incentives to deviate from the colluding strategy, so that it gets more utility for itself. However, this deviation will decrease the utility of another node (v_ℓ in the theorem) in the colluding group. Hence, as long as nodes do not fully trust each other, they are unwilling to form a colluding group.

VI. Evaluations

In Section IV, we have presented a scheme that guarantees convergence to an Admissible Strong Nash Equilibrium. In this section, we evaluate our scheme using GloMoSim. There are two sets of evaluations. The first is to illustrate the evolution of nodes' utilities and balances over time, while the second is to illustrate the effect of collusion. Our results demonstrate that our scheme is resistant to collusion.

A. Setup of Evaluation

We consider a random wireless network with 100 nodes distributed in a terrain area of 1000×1000 m. Nodes use IEEE 802.11 (at 2 Mbps) as the MAC layer protocol. The radio range is set to 140 m.

The randomly generated network topology is shown in Fig. 2. For clarity, we only include the labels of some nodes. A line between two nodes means that the two nodes are within the communication range of each other. Each node has an initial balance of 1000. We set $\epsilon = 0.001$; for each node, the cost of forwarding a unit of data is randomly chosen between 10ϵ and 100ϵ.

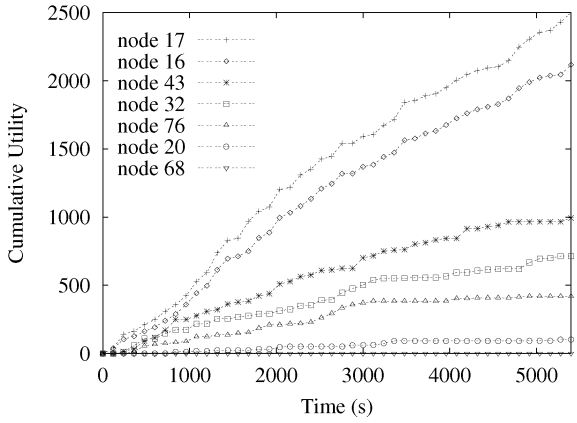


Fig. 3. Cumulative utility of nodes as a function of simulation time.

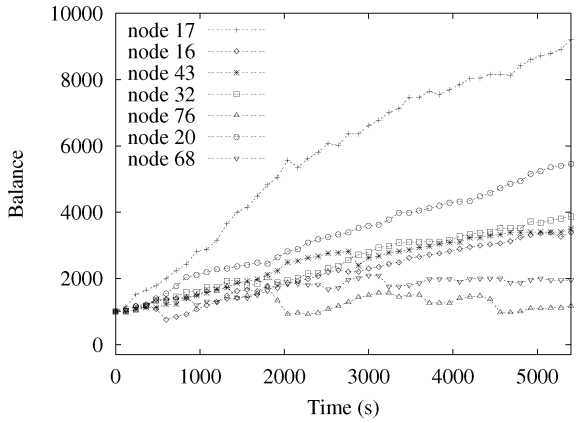


Fig. 4. Balance of nodes as a function of simulation time.

B. Evolution of Nodes' Utilities and Balances

Our first set of evaluations starts from a Nash equilibrium. The evaluation runs for 90 min, and we record the utility and balance of each node every 2 min. We generate traffic from each node according to Poisson arrival with mean time of 600 s. The destination is randomly selected from the rest of nodes. The number of units of data in each session is uniformly distributed between 1 and 1000. A node with a negative balance cannot send its own data before its balance gets positive again.

Figs. 3 and 4 show the cumulative utilities and balances of seven typical nodes during the evaluation, respectively. In general, nodes locating in the central part of the network or at a position connecting two node-dense areas (like node 16, 17, and 43) get higher cumulative utility. In contrast, nodes like 68 have much lower cumulative utility because they have less chance to earn money by forwarding others' traffic. When we compare the two figures, we can easily see that nodes' balances are not proportional to their cumulative utilities. For example, node 20 gets the second highest balance among these nodes, but it has the second least utility in the end. Node 16 collects the second largest amount of utility among these nodes, but its balance is significantly lower than node 20. This is because node 20 has a higher forwarding cost of 0.068/unit; it receives payments that are only slightly higher than its costs. In comparison, node 16 has a lower cost of 0.014/unit; the payments it receives are much more than its costs.

C. Effect of Collusion

Our second set of evaluations shows the effect of collusion. Consider a set of nodes that collude to deviate from a Nash equilibrium. (Note that, without transfer of profit, "collusion" actually means that a group of nodes deviates from the equilibrium simultaneously, in hope that each of them will benefit from the deviation.) We measure the effect of collusion by calculating the difference between each colluding node's utility and its utility in the Nash equilibrium.

First, we experiment with two different numbers of colluding nodes: 5 and 10. For each number of colluding nodes, we record the first 1000 instances of collusion that increase the total utility of the collusion set. In each run, the source node, the destination node, and the set of colluding nodes are randomly selected. Each colluding node randomly chooses one of the following actions: decreasing the claimed cost by 50%, decreasing by 20%, increasing by 20%, and increasing by 50%. For example, if a node's claimed cost is 0.1/unit in the Nash equilibrium and it increases its claimed cost by 50% in the collusion, then its claimed cost is 0.15 in the collusion. In this evaluation, there are 10 units of data in each session.

Fig. 5(a) and (b) summarize our experimental results for the effect of collusion with 5 and 10 colluding nodes, respectively. Due to limited space, we just show the first 20 records here. From experimental results, we observe that most colluding nodes do not benefit from the collusion. (In fact, most colluding nodes suffer from the collusion.) We have not found any run in which all colluding nodes benefit from the collusion. This result confirms that there is no collusion that could make all colluding nodes happy.

Fig. 6(a) and (b) demonstrate the distributions of the number of nodes that do not get more utility in collusion than in the Nash equilibrium. In these two figures, the height of each bar represents the percentage of records that have the corresponding number of colluding nodes that do not benefit from the collusion. We note that the sum of the percentages in each figure is 100%. This implies that, in all runs, we have a positive number of colluding nodes getting no more utility in collusion than in the Nash equilibrium. Therefore, there is no run in which all colluding nodes benefit.

Next, we focus on a set of four colluding nodes {16, 29, 32, 61}, which is a cutting set of the network. Suppose node 38 has 10 units of data to send to node 53. We simulate five representative cases of collusion and calculate the difference between each colluding node's utility and its utility in the Nash equilibrium. In case 1, all the colluding nodes increase their claimed cost by 50%. In case 2, all decrease by 50%. In case 3, the first half increase by 50%, and the second half decrease by 50%. In case 4, the first half decrease by 50%, and the second half decrease by 50%. In case 5, all randomly increase or decrease claimed cost.

Fig. 7 demonstrates the effect of collusion in the five cases. Just as in previous evaluations, in no case can all colluding nodes benefit from collusion.

From the above numerical results, we can see that collusion may increase average gains of a set of nodes. However, profit

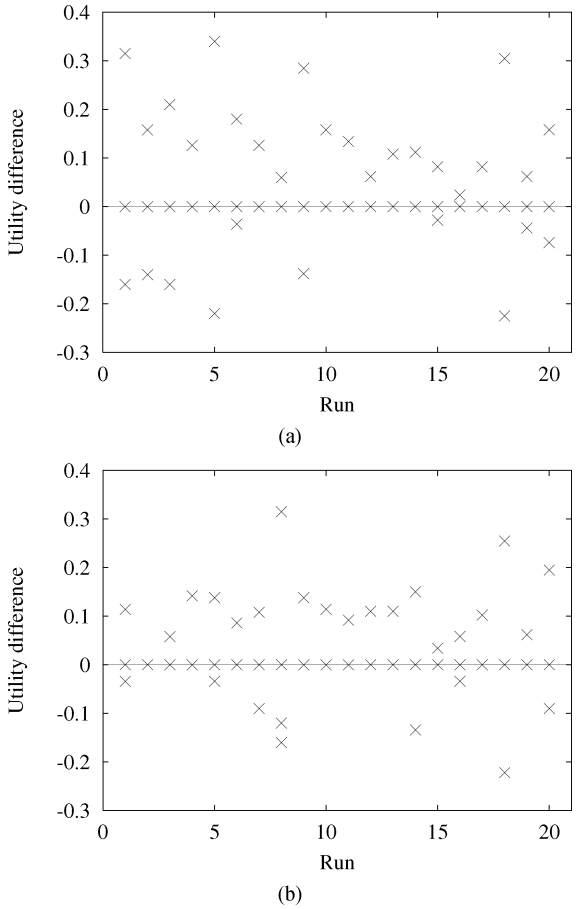


Fig. 5. Effect of collusion: Utility of each colluding node minus its utility in the Nash equilibrium. (a) 5 colluding nodes. (b) 10 colluding nodes.

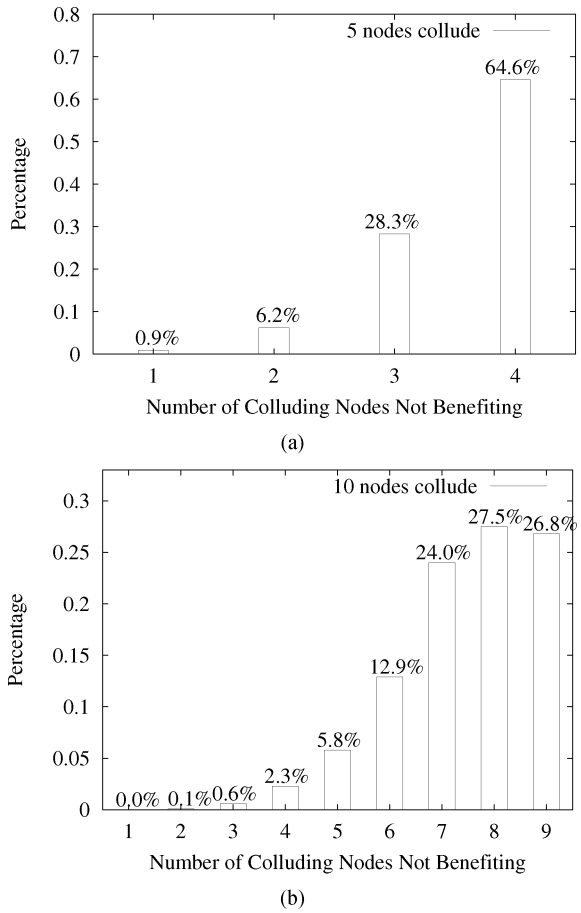


Fig. 6. Distributions of the number of colluding nodes that do not benefit from collusion. (a) 5 colluding nodes. (b) 10 colluding nodes.

transfer is required to guarantee that every colluding node benefits from the collusion. As long as profit transfer is prevented, nodes do not have incentives to collude.

D. One-Time Payment

The one-time payment used by our scheme is very small compared to the total payment. We randomly sampled 1000 sessions. The results show that the ratio between one-time payment and total payment is only 0.07%.

VII. Related Work

A considerable amount of work has been done on the incentive compatibility problems in ad hoc networks. There are two major problems: the routing problem and the packet forwarding problem. In the routing problem, we need a routing scheme that computes the lowest-cost path despite of the fact that selfish nodes can make false claims about their costs. In the packet forwarding problem, we need a protocol that stimulates selfish nodes to forward packets. We give a brief review of the existing solutions.

A. Routing in Ad Hoc Networks

Anderegg and Eidenbenz [2] were the first to address the (unicast) routing problem. Their solution Ad Hoc-VCG is based on the famous VCG mechanism, which is the standard tool to achieve strategyproofness. In [10], Eidenbenz *et al.* further considered the incentives of the service requestor and gave another VCG-based solution. A similar problem in multicast was first addressed by Wang *et al.* [34]. They showed that naive applications of VCG in the multicast scenario are not strategyproof, and then presented a solution that achieves strategyproofness without using VCG. Then, Zhong *et al.* [37] studied the combined problems of routing and packet forwarding and designed a

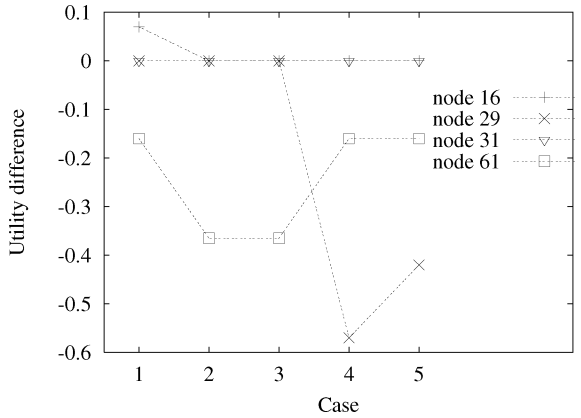


Fig. 7. Effect of collusion in five representative cases.

protocol using an integrated approach of game theory and cryptography. They showed that their solution is cooperation optimal. In [32], Wang *et al.* worked on reducing overpayments in unicast routing. Their solution, OURS, uses an elegant technique based on dummy packets and guarantees that the overpayments are low regardless of which Nash equilibrium the system converges to. As we have mentioned, all the above works on the routing problem give good solutions to stimulate each individual node to cooperate, assuming no subset of nodes would collude. However, such an assumption may not be always valid in practice. In this paper, we focus on the collusion-resistant routing problem.

An elegant result regarding collusion resistance was given by Wang and Li in [33]: While the major results of [33] also assume no collusion of nodes, they showed that dealing with collusion is hard in the sense that *True Group Strategyproofness* cannot be achieved. Here, "True Group Strategyproofness" is a new solution concept defined in [33]. Unlike the standard solution concept of Strategyproofness, True Group Strategyproofness is suitable for scenarios in which the profits gained in collusion can be transferred among colluding nodes. In comparison, in this paper we study the standard solution concepts (of Group Strategyproofness and Strong Nash Equilibrium) and propose a method to prevent transfer of profit between colluding nodes. Therefore, our work and the result in [33] are complementary to each other.

B. Packet Forwarding in Ad Hoc Networks

Incentive-compatible packet forwarding is different from, but closely related to, the routing problem. The earliest work on the packet forwarding problem was due to Marti *et al.* [22]. Their major contribution is a watchdog and a pathrater, which monitor the reputation of nodes. Similarly, Buchegger and Le Boudec's solutions [4], [5] also use an approach based on reputation. In their solutions, each node has a state machine for the reputation of other nodes; the nodes update their states according to their observations and received reports of other nodes' behavior. Generous TIT-FOR-TAT, proposed by Srinivasan *et al.* [30], is a packet forwarding strategy for selfish nodes. They showed that this strategy leads to a Nash equilibrium. Recently, Jaramillo and Srikant [19] used the theory of repeated game to study packet forwarding. Among many other interesting results, they proved that their mechanism DARWIN is optimal in their game-theoretic model. In summary, all these works provide incentives in packet forwarding using reputation systems. They are very different from our work because they require nodes to monitor others' behavior.

In this paper, we use credit, or virtual money, as compensation for participating in the game and forwarding packet. Credit was first proposed by Buttyan and Hubaux [6], [7] for the packet forwarding problem. Their solutions require each node to have a piece of tamper-proof hardware. Zhong *et al.*'s Sprite[36] is another simple credit-based solution, but it does not require tamper-proof hardware. Another solution to this problem was due to Jakobsson *et al.* [17], using a micro-payment scheme.

VIII. Conclusion and Future Work

Incentive-compatible routing is an important problem in wireless ad hoc networks. In this paper, we present a systematic study of collusion resistance in incentive-compatible routing. We focus on two standard solution concepts—Group Strategyproofness and Strong Nash Equilibrium. We show that the former is impossible to achieve and design a scheme to achieve the latter. When our scheme is used, the total payment needed is bounded. Moreover, we propose a cryptographic method that prevents profit transfer between colluding nodes, as long as they do not trust each other unconditionally. This method can be used together with our scheme that achieves Strong Nash Equilibrium. Putting the results together, we have established a theoretically sound and practically useful solution for collusion resistance in incentive-compatible routing.

Our work can be extended in several directions. The first possibility is to consider other cost models, for example, models in which a node can have different costs for different outgoing links, or models in which a node needs to determine the cost(s) with the help of its neighbors. The second possibility is to include the source and destination nodes in the routing game and investigate their incentives in the context of collusion resistance. The third possibility is to adapt our results to the scenario with probabilistic packet losses. The fourth possibility is to consider the existence of Group Strategyproof Equilibrium if admissibility is relaxed. The fifth possibility is to study where the optimal places are to put nodes, given a certain topology, if a group of nodes want to raise utility. The sixth possibility is to study how many nodes a user needs to integrate in a network in order to benefit from collusion, when profit can be transferred.

An interesting open question is related to results on a different problem: spectrum sharing. Mathur *et al.* [24], [25] found that, in the context of spectrum sharing problem, full collusion results in maximum throughput. Of course, their problem and objective are both different from ours in this paper. However, it will be interesting to consider how to integrate their results with ours to achieve maximum system performance.

We leave all the above topics to future studies.

References

[1] A. Akella, S. Seshan, R. Karp, and S. Shenker, "Selfish behavior and stability of the Internet: Game-theoretic analysis of TCP," in *Proc. SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 117–130.

[2] L. Anderegg and S. Eidenbenz, "Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *Proc. MobiCom*, San Diego, CA, Sep. 2003, pp. 245–259.

[3] N. Ben Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks," in *Proc. MobiHoc*, Annapolis, MD, Jun. 2003, pp. 13–24.

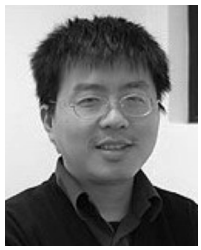
[4] S. Buchegger and J.-Y. Le Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks," in *Proc. EUROMICRO-PDP*, Canary Islands, Spain, Jan. 2002, pp. 403–410.

[5] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol (Cooperation of nodes: fairness in dynamic ad-hoc networks)," in *Proc. MobiHoc*, Lausanne, Switzerland, Jun. 2002, pp. 226–236.

[6] L. Buttyan and J. P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *Proc. MobiHoc*, Boston, MA, Aug. 2000, pp. 87–96.

[7] L. Buttyan and J. P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM J. Mobile Netw.*, vol. 8, no. 5, pp. 579–592, 2003, Special Issue on Mobile Ad Hoc Networks.

[8] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. MobiCom*, San Diego, CA, Sep. 2003, pp. 134–146.
[9] S. Eidenbenz, V. S. A. Kumar, and S. Zust, "Equilibria in topology control games for ad hoc networks," in *Proc. Joint Workshop Found. Mobile Comput.*, 2003, pp. 2–11.
[10] S. Eidenbenz, G. Resta, and P. Santi, "Commit: A sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes," presented at the IEEE IPDPS, Apr. 2005.
[11] L. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 22–26, 2001, vol. 3, pp. 1548–1557.
[12] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based mechanism for lowest-cost routing," in *Proc. 21st Symp. Principles Distrib. Comput.*, Monterey, CA, Jul. 2002, pp. 173–182.
[13] J. Feigenbaum and S. Shenker, "Distributed algorithmic mechanism design: Recent results and future directions," in *Proc. DIAL-M*, Sep. 2002, pp. 1–13.
[14] M. Felegyhazi and J.-P. Hubaux, "Wireless operators in a shared spectrum," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–11.
[15] O. Goldreich, *Foundations of Cryptography*. Cambridge, U.K.: Cambridge Univ. Press, 2001, vol. 1, Basic Tools.
[16] K. Jain and V. V. Vazirani, "Group strategyproofness and no subsidy via lp-duality," 1999.
[17] M. Jakobsson, J. P. Hubaux, and L. Buttyan, "A micropayment scheme encouraging collaboration in multi-hop cellular networks," in *Proc. Financial Crypto*, 2003, vol. 2742, pp. 15–33.
[18] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Advances in Cryptology—Eurocrypt '96*, Berlin, Germany, 1996, vol. 1070, pp. 143–154.
[19] J. Jaramillo and R. Srikant, "DARWIN: Distributed and adaptive reputation mechanism for wireless ad-hoc networks," in *Proc. MobiCom*, Montreal, QC, Canada, Sep. 2007, pp. 87–98.
[20] H. Lin, M. Chatterjee, S. K. Das, and K. Basu, "ARC: An integrated admission and rate control framework for CDMA data networks based on non-cooperative games," in *Proc. MobiCom*, San Diego, CA, Sep. 2003, pp. 326–338.
[21] S. Lindsey, K. Sivalingam, and C. Raghavendra, "Power optimization in routing protocols for wireless and mobile networks," in *Handbook of Wireless Networks and Mobile Computing*. New York: Wiley, 2002, pp. 407–423.
[22] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. MobiCom*, Boston, MA, Aug. 2000, pp. 255–265.
[23] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford, U.K.: Oxford Univ. Press, 1995.
[24] S. Mathur, L. Sankaranarayanan, and N. Mandayam, "Coalitional games in receiver cooperation for spectrum sharing," in *Proc. CISS*, Princeton, NJ, Mar. 2006, pp. 949–954.
[25] S. Mathur, L. Sankaranarayanan, and N. Mandayam, "Coalitional games in Gaussian interference channels," in *Proc. ISIT*, Seattle, WA, Jun. 2006, pp. 2210–2214.
[26] H. Moulin and S. Shenker, "Strategyproof sharing of submodular costs: Budget balance versus efficiency," *J. Econ. Theory*, vol. 18, no. 3, pp. 511–533, 2001.
[27] N. Nisan and A. Ronen, "Algorithmic mechanism design," *Games Econ. Behavior*, vol. 35, pp. 166–196, 2001.
[28] C. Papadimitriou, "Algorithms, games, and the Internet," in *Proc. 33rd Annu. Symp. Theory Comput.*, Heraklion, Crete, Greece, Jul. 2001, pp. 749–753.
[29] C. Schnorr, "Efficient signature generation for smart cards," in *Advances in Cryptology—CRYPTO '89*. New York: Springer-Verlag, 1990, pp. 239–252.
[30] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao, "Cooperation in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, vol. 2, pp. 808–817.
[31] H. Varian, "Economic mechanism design for computerized agents," in *Proc. USENIX Workshop Electron. Commerce*, 1995, vol. 1, p. 2.
[32] W. Wang, S. Eidenbez, Y. Wang, and X.-Y. Li, "OURS: Optimal unicast routing systems in non-cooperative wireless networks," in *Proc. MobiCom*, Los Angeles, CA, Sep. 2006, pp. 402–413.
[33] W. Wang and X.-Y. Li, "Low-cost routing in selfish and rational wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 596–607, May 2006.
[34] W. Wang, Li X.-Y, and Y. Wang, "Truthful multicast routing in selfish wireless networks," in *Proc. MobiCom*, New York, Sep. 2004, pp. 245–259.
[35] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. SenSys*, Los Angeles, CA, Nov. 5–7, 2003, pp. 1–13.
[36] S. Zhong, J. Chen, and Y. R. Yang, "Sprite, a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, vol. 3, pp. 1987–1997.
[37] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks – An integrated approach using game theoretical and cryptographic techniques," in *Proc. MobiCom*, Cologne, Germany, Sep. 2005, pp. 117–131.



Sheng Zhong (A'08) received the B.S. and M.E. degrees in computer science from Nanjing University, Nanjing, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from from Yale University, New Haven, CT, in 2004.

He is an Assistant Professor with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo. His research interests include economic incentives and data privacy.

Fan Wu (M'09) received the B.S. degree in computer science from Nanjing University, Nanjing, China, in 2004, and the Ph.D. degree in computer science and engineering from the University at Buffalo, The State University of New York, Buffalo, in 2009.

He is a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. His research interests include economic incentives for wireless networks and peer-to-peer computing.