On Designing Collusion-resistant Routing Schemes for Non-cooperative Wireless Ad Hoc Networks*

Sheng Zhong and Fan Wu
Computer Science and Engineering Department
The State University of New York at Buffalo
Amherst, NY 14260, U.S.A.
{szhong, fwu2}@cse.buffalo.edu

ABSTRACT

In wireless ad hoc networks, routing requires 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 resistance in incentive-compatible routing schemes. 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. In addition, we give a cryptographic method that prevents profit transfer between 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.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design–Wireless communication

General Terms

Algorithms, Design, Economics, Security, Theory

Keywords

Collusion, routing, wireless ad hoc networks

*Sheng Zhong was supported in part by NSF CNS-0524030. Fan Wu was supported by NSF CNS-0524030.

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.

1. INTRODUCTION

With the increasing popularity of portable devices, 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 this network. For example, if we want to route packets through the lowest-cost path, then we need the information from each node about its cost for forwarding packets. In civilian ad hoc networks, nodes often belong to different nodes and have their own interests; they may not want to behave cooperatively. Consequently, it is highly important to provide incentives for nodes to cooperate.

The problem of incentive-compatible routing has received much attention [9, 25–27, 29]. Nevertheless, most existing solutions focus on the economic incentives of each *individual node*, assuming that no subset of nodes would collude. In many practical scenarios, this assumption is not always valid. 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 [26]. 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 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? These standard solution concepts are applicable in practice if there is no profit transfer between colluding nodes. Can we make sure that there is no such profit transfer? Is there a method to prevent such profit transfer? The objective of this paper is to present a systematic study of collusion resistance to address the above 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 show that we can prevent profit transfer between colluding nodes, as long as they do not fully trust each other unconditionally.[1] Thus, the scheme we design to achieve Strong Nash Equilibrium is very practical. In particular, we show 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.

- Finally, we evaluate our solution using extensive experiments. Simulations on randomly generated networks and on a snapshot network demonstrate that our scheme is resistant to collusion. Measurements of the overheads of our solution show that it is very efficient.

The rest of the paper is organized as follows. Section 2 presents the technical preliminaries. The impossibility of achieving Group Strategyproofness is proved in Section 3 and the scheme to achieve Strong Nash Equilibrium is given in Section 4. Section 5 is dedicated to our method to prevent profit transfer between colluding nodes. Section 6 gives our evaluation results. In Section 7, we discuss related work. We conclude the paper in Section 8.

2. 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 bi-connected.

For each node $v_i \in V$, there is a cost $c_i \in \mathbb{R}^+$ for sending a unit of data to its neighbors. We adopt the assumption [25] that a node cannot use different power levels to send data to different neighbors, so that the cost is fixed for each node. Further, each node v_i knows the cost c_i.

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 $V-\{S, D\}$. In this game, each player node v_i chooses an action based on its own cost: $a_i = \mathcal{A}_i(c_i)$. Denote by a the profile of all players' actions: $a = (a_i)_{v_i \in V-\{S,D\}}$. 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. Techniques from [29] can be useful for those issues.) 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:

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

[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.

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.

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 1. *(Group Strategyproof Equilibrium [14,20]) 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,

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

DEFINITION 2. *(Strong Nash Equilibrium [19]) 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 3. *(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 3 and 4: 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? Our answer is that the former is impossible, while the latter can be achieved.

3. 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 4. *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. So, for an arbitrary unicast routing game, we show that any Strategyproof Equilibrium a^* is not Group Strategyproof if it is admissible.

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

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

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

LEMMA 5. *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

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

PROOF. Since a^* is admissible, when c is the cost profile, clearly we have

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

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

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

Furthermore, from (1) we know that

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

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. For all $v_i \in C$, $i \neq i_0$, clearly $\sigma_i(a^*_C, a^*_{\overline{C}}) = 0$. By Lemma 5, this implies that

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

Next, we combine (3) with (2). Since a^* is Strategyproof, considering the scenario in which $(c'_C, c_{\overline{C}})$ is the cost profile, we get:[2]

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

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

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

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

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

[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 Lemma 5 it suffices to have (4).

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

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

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

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

Since $\sigma_{i_0}(a^*_C, a^*_{\overline{C}}) = 1$, using (8) and Lemma 5 we get that, when c is the cost profile,

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

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

4. SCHEME ACHIEVING STRONG NASH EQUILIBRIUM

In Section 3, 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.

The key idea of our design is discretization of costs. In practice, the cost c_i of each node has a finite precision. So, 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 necessary 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 the above are only some intuitive thoughts behind our design, which are not completely precise. For precise analysis, see the theorems, lemmas, and proofs we present below.

Suppose S wants to send n units of data to D.

- Each player node v_i sends $a_i \in \mathbb{R}^+ \cup \{0\}$ to S, which is its claimed cost. (So, *each player's action is its claimed cost.*)
- S chooses the lowest-cost path (LCP) to D in the graph with the claimed costs. If there is a tie, S breaks the tie according to the lexicographical order. This is the path for forwarding the data from S to D. Each node v_i in this selected path is paid $p_i(a) = a_i$ for each unit of data. (Recall that $LCP()$ denotes "the lowest-cost path". Hereafter, whenever there are more than one lowest-cost paths, $LCP()$ always refers to the one selected using the above tie-breaking rule.)
- In addition, each node v_i in the selected path receives a one-time payment:

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

  In the above, $\max_{v_i \in LCP(S,D,(a_i', a^*_{\overline{\{i\}}}))} a_i'$ 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 selected path. If node v_i is not in the selected path, it receives a one-time payment:

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

Figure 1: Scheme for Achieving Strong Nash Equilibrium.

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 two major results: (1) there exists a Nash Equilibrium; (2) all Nash Equilibria are admissible Strong Nash Equilibria. We start with proving the first result.

THEOREM 6. *If the above scheme 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 $v_i \notin LCP(S, D, c)$, if $a_i^* > 0$ and $LCP(S, D, (a_i^* - \epsilon, a^*_{\overline{\{i\}}})) = LCP(S, D, a^*)$, we decrease a_i^* by ϵ; otherwise, keep the value of a_i^*. For each player node $v_i \in LCP(S, D, c)$, if $LCP(S, D, (a_i^* + \epsilon, a^*_{\overline{\{i\}}})) = LCP(S, D, a^*)$, 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, $u_i(a_i^*, a^*_{\overline{\{i\}}}) \geq u_i(a_i, a^*_{\overline{\{i\}}})$. We distinguish two cases:

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

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

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

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

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

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

Case B: $v_i \in LCP(S, D, c) = LCP(S, D, a^*)$. Then,

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

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

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

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

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

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

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

□

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

LEMMA 7. *If the above scheme is used, then for each Nash equilibrium a^*, $LCP(S,D,c) = LCP(S,D,a^*)$. 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 $LCP(S,D,c) \neq LCP(S,D,a^*)$. We distinguish two cases.

Case A: There exists v_i such that $v_i \in LCP(S,D,c)$, $v_i \notin LCP(S,D,a^*)$, $a_i^* > c_i$. 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 $v_i \in LCP(S,D,(c_i, a^*_{\{i\}}))$,

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

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

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

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

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

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

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

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

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

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

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

we get that

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

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

LEMMA 8. *If the above scheme is used, then for all Nash Equilibrium a^*, we have that $a_i^* \geq c_i \Leftrightarrow v_i \in LCP(S,D,c)$ and that $a_i^* \leq c_i \Leftrightarrow v_i \notin LCP(S,D,c)$.*

PROOF. We only need to show that $v_i \in LCP(S,D,c) \Rightarrow a_i^* \geq c_i$ and that $v_i \notin LCP(S,D,c) \Rightarrow a_i^* \geq c_i$, which are equivalent to this lemma.

First, we prove $v_i \in LCP(S,D,c) \Rightarrow a_i^* \geq c_i$ by contradiction. Suppose that there exists $v_i \in LCP(S,D,c)$, such that $a_i^* < c_i$. Since $LCP(S,D,c) = LCP(S,D,a^*)$ (by Lemma 7), v_i's equilibrium utility is

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

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 $v_i \notin LCP(S,D,c) \Rightarrow a_i^* \leq c_i$, also by contradiction. Suppose that there exists $v_i \notin LCP(S,D,c)$, such that $a_i^* > c_i$. Since $LCP(S,D,c) = LCP(S,D,a^*)$ (by

Lemma 7), 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 $v_i \in LCP(S, D, (c_i, a^*_{\overline{\{i\}}}))$, 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 $v_i \notin LCP(S, D, (c_i, a^*_{\overline{\{i\}}}))$, then

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

This completes the proof. □

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

THEOREM 9. *If the above scheme is used, then all Nash Equilibria are admissible Strong Nash Equilibria.*

PROOF. (Sketch) It is clear from Lemma 7 and Lemma 8 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 $v_i \in LCP(S, D, (a_C^*, a^*_{\overline{C}}))$, then $v_i \in LCP(S, D, (a_C, a_C^*))$. Suppose that there exists v_i, $v_i \in LCP(S, D, (a_C^*, a^*_{\overline{C}}))$, $v_i \notin LCP(S, D, (a_C, a^*_{\overline{C}}))$. Assume that the tie breaking rule prefers $LCP(S, D, (a_C^*, a^*_{\overline{C}}))$ to $LCP(S, D, (a_C, a^*_{\overline{C}}))$ when their claimed costs are equal. (We have a similar proof when the tie breaking rule prefers $LCP(S, D, (a_i, a^*_{\overline{C}}))$.) Then, we have

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

Since

$$\begin{aligned} & \sum_{v_i \in LCP(S,D,(a_C, a^*_{\overline{C}})), v_i \notin LCP(S,D,(a_C^*, a^*_{\overline{C}}))} a_i^* \\ \geq \; & \sum_{v_i \in LCP(S,D,(a_C^*, a^*_{\overline{C}})), v_i \notin LCP(S,D,(a_C, a^*_{\overline{C}}))} a_i^*, \end{aligned}$$

we have

$$\begin{aligned} & \sum_{v_i \in LCP(S,D,(a_C, a^*_{\overline{C}})), v_i \notin LCP(S,D,(a_C^*, a^*_{\overline{C}})), v_i \in C} (a_i - a_i^*) \\ < \; & \sum_{v_i \in LCP(S,D,(a_C^*, a^*_{\overline{C}})), v_i \notin LCP(S,D,(a_C, a^*_{\overline{C}})), v_i \in C} (a_i - a_i^*). \end{aligned} \tag{11}$$

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

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

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

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

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

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

$$v_i \in C \wedge v_i \in LCP(a_C^*, a^*_{\overline{C}}) \Leftrightarrow a_i^* < a_i;$$

$$v_i \in C \wedge v_i \notin LCP(a_C^*, a^*_{\overline{C}}) \Leftrightarrow a_i^* > a_i.$$

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

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

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

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

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

5. PREVENTING PROFIT TRANSFER BETWEEN COLLUDING NODES

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 widely applicable in practice, we provide 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. If v_1 takes action a_1 and v_2 takes action a_2, then v_1 has an increases of 10 in its utility but v_2 has a decrease of 5 in its utility. So, 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 that v_2 can convince v_1 about its action. Our method makes 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. Therefore, when our method is used, v_1 should not trust v_2's claim and should not be willing 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 which 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 develop a new cryptographic technique called *restricted verifier signature*.[4]

When the source node S makes a payment to a player node, it signs its payment using our restricted verifier signature. 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, our 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 give the details of our restricted verifier signature scheme, which is based on the well known Shnorr signature [23], and then briefly analyze it.

5.1 The Scheme of Restricted Verifier Signature

Let $\mathcal{P}$ and $\mathcal{Q}$ be two large primes such that $\mathcal{P} = 2\mathcal{Q} + 1$. Denote by $\mathbb{Z}_{\mathcal{P}}^*$ the multiplicative group mod $\mathcal{P}$; denote by $\mathbb{G}_{\mathcal{Q}}$ the subgroup of quadratic residues. Clearly, $|\mathbb{G}_{\mathcal{Q}}| = \mathcal{Q}$. Denote by $\mathbb{Z}_{\mathcal{Q}}$ the additive group mod $\mathcal{Q}$. Let g be a generator of $\mathbb{G}_{\mathcal{Q}}$. The above parameters $\mathcal{P}$, $\mathcal{Q}$, and g are all public.

Suppose that each node v_i has a private key $x_i \in \mathbb{Z}_{\mathcal{Q}}$ and a public key $y_i = g^{x_i}$. Assume that there is a central bank with private key $x_B \in \mathbb{Z}_{\mathcal{Q}}$ and public key $y_B = g^{x_B}$.

Suppose that $\mathsf{Sign}()$ is a standard digital signing algorithm and that $\mathsf{Verify}()$ is the corresponding verification algorithm. Suppose that $H()$ is a cryptographic hash function.

To sign a message m that can only be verified by v_j and the bank, node v_i computes:

$$\begin{aligned}
\overline{m}_1 &= g^{r_1} \pmod{\mathcal{P}},\\
\overline{m}_2 &= y_B^{m+r_1} \pmod{\mathcal{P}},\\
\overline{m}_3 &= \mathsf{Sign}_{x_i}(\overline{m}_1, \overline{m}_2),\\
\overline{m}_4 &= g^{r_2} y_j^{r_3} \pmod{\mathcal{P}},\\
\overline{m}_5 &= y_B^{r_2} \pmod{\mathcal{P}},\\
\overline{m}_6 &= r_1 \cdot H(\overline{m}_1, \overline{m}_2, \overline{m}_3, \overline{m}_4, \overline{m}_5) + r_2 \pmod{\mathcal{Q}},
\end{aligned}$$

where r_1, r_2, r_3 are picked uniformly and independently from $\mathbb{Z}_{\mathcal{Q}}$. The signature is $(\overline{m}_1, \overline{m}_2, \overline{m}_3, \overline{m}_4, \overline{m}_5, \overline{m}_6, r_3)$.

To verify the above signature, node v_j checks that

$$g^{\overline{m}_6} y_j^{r_3} = \overline{m}_1^{H(\overline{m}_1, \overline{m}_2, \overline{m}_3, \overline{m}_4, \overline{m}_5)} \cdot \overline{m}_4 \pmod{\mathcal{P}}, \qquad (13)$$

[3]One may suggest that v_1 should transfer 7 to v_2 only after the path and payments turn out to be those produced by collusion. 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, v_1 can decline to transfer anything to v_2 when v_2 takes action a_2.

[4]Our restricted verifier signature is closely related to the well known *designated verifier signature* and *multiple designated verifier signature* [16], but is different. 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 our restricted verifier signature scheme.

[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.

$$y_B^{\overline{m}_6} = (\overline{m}_2/y_B^m)^{H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)} \cdot \overline{m}_5 \pmod{\mathcal{P}}, \quad (14)$$

and that

$$\mathsf{Verify}_{y_i}((\overline{m}_1,\overline{m}_2),\overline{m}_3) = \mathsf{Accept}. \quad (15)$$

When the bank needs to verify this signature, it only needs to check (15) and that

$$\overline{m}_2 = y_B^m \cdot \overline{m}_1^{x_B} \pmod{\mathcal{P}}. \quad (16)$$

5.2 Analysis

Now we analyze our restricted verifier signature scheme. We prove four properties of this scheme. First, we show that our scheme is correct in the sense that a valid signature can always be verified by the node v_j and the central bank. Second, we show that any signature accepted by v_j contains a valid payment that will be honored by the bank. Third, we show that the signature cannot be forged. Finally, we show that any party other than v_j and the bank cannot verify the signature. These four properties make it possible for us to prevent profit transfer.

THEOREM 10. *(Correctness of Signature) In the above signature scheme, a valid signature can always be verified by v_j and the bank.*

PROOF. It is easy to see that (15) and (16) hold. So we only need to show (13) and (14):

$$\begin{aligned} g^{\overline{m}_6} y_j^{r_3} &= g^{r_1 \cdot H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)+r_2} y_j^{r_3} \\ &= (g_1^r)^{H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)} g^{r_2} y_j^{r_3} \\ &= \overline{m}_1^{H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)} \cdot \overline{m}_4 \pmod{\mathcal{P}}; \end{aligned}$$

$$\begin{aligned} y_B^{\overline{m}_6} &= y_B^{r_1 \cdot H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)+r_2} \\ &= (y_B^{r_1})^{H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)} y_B^{r_2} \\ &= (\overline{m}_2/y_B^m)^{H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5)} \cdot \overline{m}_5 \pmod{\mathcal{P}}. \end{aligned}$$

□

THEOREM 11. *(Binding Property) In the random oracle model, if node v_j accepts $\overline{m}$ as a signature on message m, then there exists $r_1 \in \mathbb{Z}_{\mathcal{Q}}$ such that $(\overline{m}_1,\overline{m}_2) = (g^{r_1}, y_B^{m+r_1})$.*

PROOF. (Sketch) When node v_j accepts $\overline{m}$ as a signature on message m, clearly (13) and (14) hold. Note that $H()$ is a cryptographic hash function. Using the random oracle model, we can assign random values as the output of this hash function, and (13) and (14) should still hold. From (13) we get

$$g^{\overline{m}_{6,1}} y_j^{r_3} = \overline{m}_1^{H_1} \cdot \overline{m}_4 \pmod{\mathcal{P}}, \quad (17)$$

$$g^{\overline{m}_{6,2}} y_j^{r_3} = \overline{m}_1^{H_2} \cdot \overline{m}_4 \pmod{\mathcal{P}}, \quad (18)$$

where H_1, H_2 are two random hash values, $\overline{m}_{6,1}, \overline{m}_{6,2}$ are the corresponding values of $\overline{m}_6$. Combining (17) and (18), we get

$$g^{\overline{m}_{6,1}-\overline{m}_{6,2}} = \overline{m}_1^{H_1-H_2} \pmod{\mathcal{P}}.$$

Let $r_1 = (\overline{m}_{6,1} - \overline{m}_{6,2})/(H_1 - H_2)$. The above equation is equivalent to $\overline{m}_1 = g^{r_1}$. Similarly we can also get from (14): $\overline{m}_2 = y_B^{m+r_1}$. □

THEOREM 12. *(Unforgeability) An adversary knowing neither x_i nor x_j cannot forge a signature satisfying (13)(14)(15).*

THEOREM 13. *(Restriction of Verifiability) From a valid signature $\overline{m}$ on message m, for an arbitrary different message $m' \neq m$, node v_j can compute a string $\tilde{m}'$ that is computationally indistinguishable[6] from a valid signature on m' by any party other than the bank and v_i.*

PROOF. Node v_j can compute $\tilde{m}'$ as follows: define $\tilde{m}'_1 = \overline{m}_1$, $\tilde{m}'_2 = \overline{m}_2$, $\tilde{m}'_3 = \overline{m}_3$, $\tilde{m}'_4 = \overline{m}_4$, $\tilde{m}'_5 = \overline{m}_5$. Then, compute

$$\tilde{m}'_6 = \overline{m}_6 + (m - m')H(\overline{m}_1,\overline{m}_2,\overline{m}_3,\overline{m}_4,\overline{m}_5) \pmod{\mathcal{Q}};$$

$$r'_3 = r_3 + (\overline{m}_6 - \tilde{m}'_6)x_j^{-1} \pmod{\mathcal{Q}}.$$

The string $\tilde{m}'$ is defined as $(\tilde{m}'_1, \tilde{m}'_2, \tilde{m}'_3, \tilde{m}'_4, \tilde{m}'_5, \tilde{m}'_6, r'_3)$. □

6. EVALUATIONS

In Section 4, we have presented a scheme that guarantees convergence to a Strong Nash Equilibrium. Using GloMoSim, we evaluate this scheme on two networks. The first is randomly generated and the second is based on a snapshot of an urban area in New York City. For both networks we demonstrate that our scheme is resistant to collusion. Furthermore, to evaluate the efficiency of our entire solution, we also measure the computational overhead of our restricted verifier signature scheme, which is computationally the most expensive part of our solution.

6.1 Evaluation on Random Wireless Network

We consider a random wireless network with 100 nodes distributed in a terrain area of 3000 by 3000 meters. Nodes use IEEE 802.11 (at 2Mbps) as the MAC layer protocol. The radio range is set to 422.757 meters.

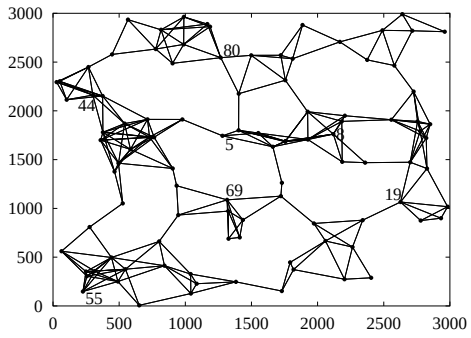


Figure 2: Topology of the Random Generated Network.

The randomly generated network topology is shown in Figure 2. For clarity, we only include the labels of a few nodes. A line between two nodes means that the two nodes are in 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.

We have two evaluations done on the random wireless network. The first is to illustrate the evolution of nodes' utilities and balances over time, while the second is to illustrate the effect of collusion.

[6]Here being *computationally indistinguishable* means that the string $\tilde{m}'$ cannot be distinguished from a valid signature on m' by any polynomial-time adversary. See [13] for the precise definition.

Our first evaluation starts from a Nash Equilibrium. The evaluation runs for 90 minutes and we observe the utility and balance of each node every 2 minutes. We generate traffic from each node according to Poisson arrival with mean time of 600 seconds. 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.

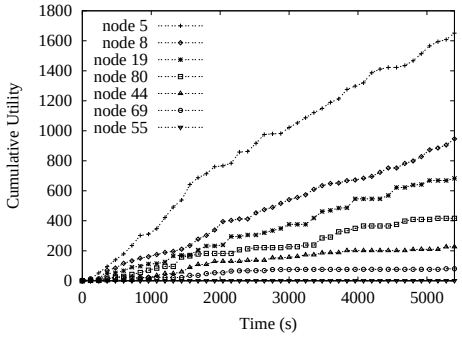


Figure 3: Cumulative Utility of Nodes as a Function of Simulation Time.

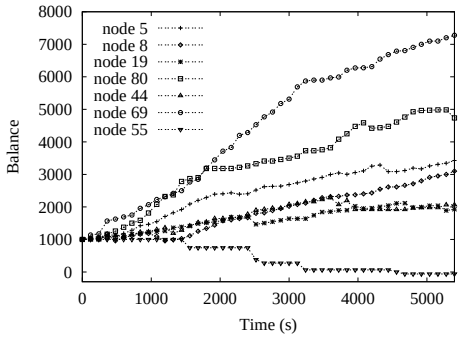
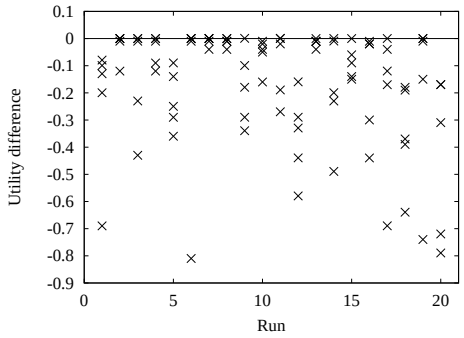


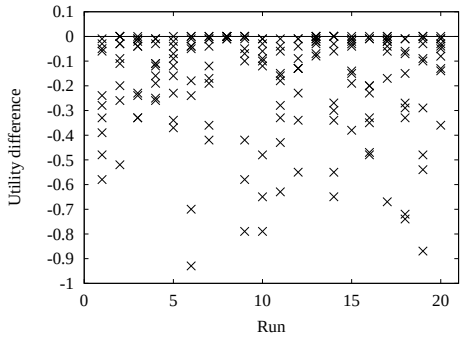
Figure 4: Balance of Nodes as a Function of Simulation Time.

Figure 3 and Figure 4 show the cumulative utilities and balances of seven typical nodes during the evaluation, respectively. Generally, nodes locating in the central part of the network or at a position connecting two node-dense areas (like node 69 and node 80) get higher balances. In contrast, nodes like 55 have much lower balances, 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 69 gets the highest balance among these nodes, but it has the second least utility in the end. Node 5 collects the largest amount of utility among these nodes, but its balance is significantly lower than node 69 and node 80. This is because node 69 has a high forwarding cost of 0.093/unit; it receives payments which are only slightly higher than its costs. In comparison, node 5 has a low cost of 0.024/unit; the payments it receives are much more than its costs.

Our second evaluation shows the effect of collusion. Consider a set of nodes that collude to deviate from a Nash



(a) 5 colluding nodes.



(b) 10 colluding nodes.

Figure 5: Effect of Collusion: Utility of Each Colluding Node Minus Its Utility in the Nash Equilibrium.

Equilibrium. (Note that, without transfer of profit, "collusion" actually means that a group of nodes deviate 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. We experiment with two different numbers of colluding nodes: 5 and 10. For each number of colluding nodes, we have 20 runs of the experiment. In each run, the source node, the destination node, and the set of colluding nodes are randomly selected. The action of each colluding node is randomly chosen from four possibilities: decreasing its 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.

Figure 5 summarizes our experimental results for the effect of collusion with 5 and 10 colluding nodes, respectively. From Figure 5, we can see 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.

6.2 Network Based on a Snapshot

In this section, we use a snapshot of a New York City urban area (see Figure 6). The terrain area is about 750m × 750m. We assume that 10% of the automobiles and 20%

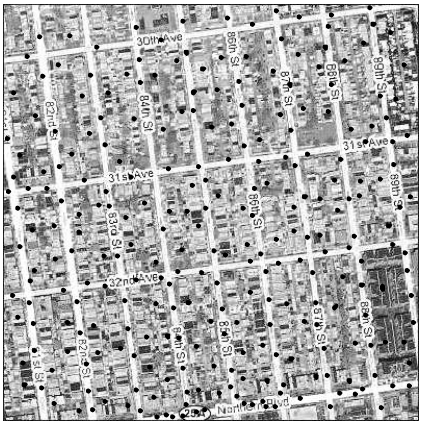
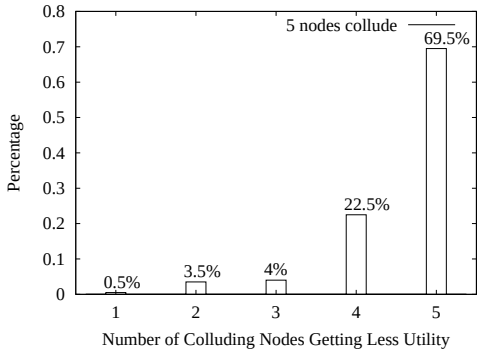
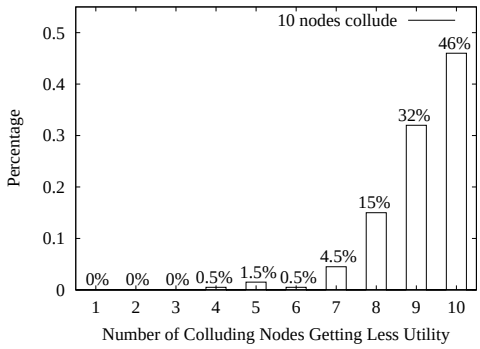


Figure 6: Snapshot of New York City Urban Area and Locations of Wireless Devices.

of the buildings are equipped with wireless devices. Thus, the total number of wireless devices in this network is 267. The radio range is set to 75m. In Figure 6, each dot denotes a wireless device. For each wireless device, the cost of forwarding a unit of data is randomly selected between 10ϵ and 100ϵ ($\epsilon = 0.001$). We evaluate the effect of collusion using the same method as on the randomly generated network, except that we have 200 runs of the experiment for each number of colluding nodes.



(a) 5 colluding nodes.



(b) 10 colluding nodes.

Figure 7: Distributions of the Number of Nodes Getting Less Utility in Collusion.

Figures 7 (a) and (b) demonstrate the distributions of the number of nodes that get less utility in collusion than in the Nash Equilibrium. In these two figures, the height of each bar represents the percentage of runs that have the corresponding number of colluding nodes suffer 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 less utility in collusion than in the Nash Equilibrium. Therefore, there is no run in which all colluding nodes benefit.

6.3 Efficiency of Restricted Verifier Signature

As we have mentioned, our scheme for Strong Nash Equilibrium can be used together with our restricted verifier signature to make a practical solution. In this practical solution, most computational overheads come from our restricted verifier signature. Consequently, we carry out experiments to measure the efficiency of our restricted verifier signature scheme.

We implement our restricted verifier signature scheme using Crypto++. The standard digital signing/verification algorithm we use are DSA, RSA (as specified in ANSI X9.31), and Elliptic Curve DSA (ECDSA; as specified in ANSI X9.62). We measure the computational overheads using three different hash functions: MD5, SHA-1, and SHA-256. The evaluation is performed on a laptop with 1.4GHz Centrino CPU and 768MB memory. Table 1 lists the computational overheads with two different key lengths: 1024 bits and 2048 bits. In the table, each entry has three numbers. The first number is the signing time; the second is the node's verification time; and the third is the bank's verification time. We can see that the choice of hash function has little influence on the overheads while the choice of standard signature scheme has some influence. However, regardless of which standard signature scheme is used, our restricted verifier signature scheme remains very efficient. For the key length of 1024 bits, all operations are guaranteed to be finished in less than 0.1 second. (In our work, it is as important to have efficient signing as to have efficient verification, since our routing scheme requires the source node to generate a signature for each intermediate node in each session.)

7. 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. Below we give a brief review of the existing solutions.

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 [9], 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. [27]. 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.

	Keysize	MD5	SHA-1	SHA256
DSA	1024 bits	0.052s/0.058s/0.034s	0.052s/0.058s/0.035s	0.053s/0.063s/0.035s
	2048 bits	0.315s/0.333s/0.211s	0.315s/0.337s/0.210s	0.318s/0.348s/0.211s
ECDSA	1024 bits	0.047s/0.048s/0.026s	0.047s/0.049s/0.025s	0.047s/0.053s/0.026s
	2048 bits	0.272s/0.249s/0.125s	0.272s/0.255s/0.126s	0.272s/0.262s/0.126s
RSA	1024 bits	0.049s/0.044s/0.021s	0.047s/0.046s/0.021s	0.048s/0.049s/0.021s
	2048 bits	0.283s/0.250s/0.129s	0.281s/0.250s/0.124s	0.283s/0.263s/0.125s

Table 1: Computation Overhead (Signing time/Verifying time by node/Verifying time by bank).

Then, Zhong et al. [29] studied the combined problems of routing and packet forwarding and designed a protocol using an integrated approach of game theory and cryptography. They showed that their solution is cooperation optimal. In [25], Wang et al. worked on reducing over-payments in unicast routing. Their solution OURS uses an elegant technique based on dummy packets and guarantees that the over-payments are low regardless of which Nash Equilibrium the system converges to.

As we have mentioned, all the above work on the routing problem focus on the incentives of each individual node. In other words, it is assumed that nodes do not collude. Clearly, this assumption is not always valid in practice. An elegant result regarding collusion resistance was given by Wang and Li in [26]: While the major results of [26] 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 [26]. 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 provide a method to prevent transfer of profit between colluding nodes. So, our work and the result in [26] are complementary to each other.

Packet Forwarding in Ad Hoc Networks.

The earliest work on the packet forwarding problem was due to Marti et al. [18]. 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. [24], is a packet-forwarding strategy for selfish nodes. They showed that this strategy leads to a Nash Equilibrium.

Buttyan and Hubaux [6, 7] proposed to use credit or virtual money for the packet forwarding problem. Their solutions require each node to have a piece of tamper-proof hardware. Zhong et al.'s Sprite [28] is another simple credit-based solution but it does not require tamper-proof hardware. Ben Salem et al. [3] addresses the packet forwarding problem in multi-hop cellular networks, using a protocol based on symmetric key cryptography. Another solution to this problem was due to Jakobsson et al. [15], using a micro-payment scheme.

Other Work in Networking.

In addition to the two problems we discuss above, There are many other problems in computer networks that are addressed using game theory [1, 10, 11, 21, 22]. Examples include Eidenbenz's topology control game for ad hoc networks [8], Lin et al.'s admission and rate control for CDMA networks [17], and Felegyhazi and Hubaux's spectrum sharing for wireless operators [12].

8. 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. Moreover, we give 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. One 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. Another possibility is to include the source and destination nodes in the routing game and investigate their incentives in the context of collusion resistance. Yet another possibility is to adapt our results to the scenario with probabilistic packet losses. We leave these topics to future study.

9. REFERENCES

[1] A. Akella, S. Seshan, R. Karp, and S. Shenker. Selfish behavior and stability of the internet: Game-theoretic analysis of TCP. In Proceedings of the Special Interest Group on Data Communication (SIGCOMM), Pittsburgh, PA, August 2002.

[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 Proceedings of the Ninth International Conference on Mobile Computing and Networking (MobiCom), San Diego, CA, Sep. 2003.

[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

Proceedings of the Fourth ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, MD, Jun. 2003.

[4] S. Buchegger and J.-Y. Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP), Canary Islands, Spain, Jan. 2002.

[5] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the CONFIDANT protocol (Cooperation of nodes: fairness in dynamic ad-hoc networks). In Proceedings of the Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Lausanne, Switzerland, Jun. 2002.

[6] L. Buttyan and J. P. Hubaux. Enforcing service availability in mobile ad-hoc WANs. In Proceedings of the First ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc), Boston, Massachusetts, Aug. 2000.

[7] L. Buttyan and J. P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad Hoc Networks, summer 2002.

[8] S. Eidenbenz, V. S. A. Kumar, and S. Zust. Equilibria in topology control games for ad hoc networks. In Proceedings of the 2003 Joint Workshop on Foundations of Mobile Computing, pages 2-11, 2003.

[9] S. Eidenbenz, G. Resta, and P. Santi. Commit: A sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes. In Proceedings of 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (IPDPS), Apr. 2005.

[10] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In Proceedings of the 21st Symposium on Principles of Distributed Computing, pages 173-182, Monterey, CA, Jul. 2002.

[11] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In Proceedings of the Sixth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), pages 1-13. ACM Press, Sep. 2002.

[12] M. Felegyhazi and J.-P. Hubaux. Wireless operators in a shared spectrum. In Proceedings of the 25th Conference on Computer Communications (INFOCOM), Barcelona, Spain, Apr. 2006.

[13] O. Goldreich. Foundations of Cryptography: Volume 1, Basic Tools. Cambridge University Press, Aug. 2001.

[14] K. Jain and V. V. Vazirani. Group strategyproofness and no subsidy via lp-duality, 2002.

[15] M. Jakobsson, J. P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In Proceedings of Financial Crypto 2003, volume 2742, pages 15-33, 2003.

[16] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In Advances in Cryptology - Eurocrypt '96, volume 1070, pages 143-154, Berlin, 1996.

[17] 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 Proceedings of the Ninth International Conference on Mobile Computing and Networking (MobiCom), pages 326-338, San Diego, CA, Sep. 2003.

[18] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In Proceedings of the Sixth International Conference on Mobile Computing and Networking (MobiCom), Boston, MA, Aug. 2000.

[19] A. Mas-Colell, M. D. Whinston, and J. R. Green. Microeconomic Theory. Oxford Press, 1995.

[20] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. In Economic Theory, 2002.

[21] N. Nisan and A. Ronen. Algorithmic mechanism design. Games and Economic Behavior, 35:166-196, 2001.

[22] C. Papadimitriou. Algorithms, games, and the Internet. In Proceedings of the 33rd Annual Symposium on Theory of Computing, pages 749-753, Heraklion, Crete, Greece, Jul. 2001.

[23] C. Schnorr. Efficient signature generation for smart cards. In Advances in Cryptology - CRYPTO '89, pages 239-252. Springer-Verlag, 1990.

[24] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao. Cooperation in wireless ad hoc networks. In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), San Francisco, CA, Apr. 2003.

[25] W. Wang, S. Eidenbez, Y. Wang, and X.-Y. Li. OURS: Optimal unicast routing systems in non-cooperative wireless networks. In Proceedings of The Twelfth International Conference on Mobile Computing and Networking (MobiCom), Los Angeles, CA, Sep. 2006.

[26] W. Wang and X.-Y. Li. Low-cost routing in selfish and rational wireless ad hoc networks. IEEE Transactions on Mobile Computing, 5(5):596-607, 2006.

[27] W. Wang, X.-Y. Li, and Y. Wang. Truthful multicast routing in selfish wireless networks. In Proceedings of the Tenth International Conference on Mobile Computing and Networking (MobiCom), pages 245-259, New York, NY, USA, Sep. 2004. ACM Press.

[28] S. Zhong, J. Chen, and Y. R. Yang. Sprite, a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), San Francisco, CA, Apr. 2003.

[29] 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 Proceedings of The 11th International Conference on Mobile Computing and Networking (MobiCom), Cologne, Germany, Sep. 2005.