This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2008 proceedings.

A Novel Incentive-Based and Hardware-Independent Cooperation Mechanism for MANETs

Hamed Janzadeh
Computer Eng. Department
Amirkabir University of Tech.
Tehran, IRAN
janzadeh@aut.ac.ir

Kaveh Fayazbakhsh
Computer Eng. Department
Amirkabir University of Tech.
Tehran, IRAN
kfayazbakhsh@aut.ac.ir

Bahador Bakhshi
Computer Eng. Department
Amirkabir University of Tech.
Tehran, IRAN
bbakhshi@aut.ac.ir

Mehdi Dehghan
Computer Eng. Department
Amirkabir University of Tech.
Tehran, IRAN
dehghan@aut.ac.ir

***Abstract*—One of the most challenging issues in mobile ad-hoc networks (MANETs) arena, which consist of autonomous and self interested nodes, is the problem of providing incentives for nodes to cooperate in forwarding network packets. In this paper, we propose Express as a cooperation mechanism which is both efficient in computations imposed on mobile nodes and secure against deceptive threats by nodes. Using Express, network nodes gain credit for participating in forwarding network packets. Recent works utilize digital signatures to provide security against cheating actions by nodes. Express substitutes hash operations for digital signature operations as much as possible, which consequently reduces computational overhead to a great extent. Setting credits and penalties through this mechanism, each node beneficially tends to adopt cooperative behavior.**

Keywords-MANET; cooperation mechanism; hash chain; game theory.

I. INTRODUCTION

Rapid expanding range of capabilities and various uses of mobile computing devices have made mobile ad-hoc networks (MANETs) of great interest to both researchers and commercial developers. In recent years, extensive use of mobile devices in such applications as access to the Internet when there is not any access point within the range of mobile device has brought MANET as a business platform to researchers' attention [1]. It seems probable that MANET nodes do not belong to the same authority; but are autonomous entities which manage their own resources like battery power and available data transmission bandwidth [2]. Critical value of MANET nodes battery power and available transmission bandwidth are major motivation for not forwarding other nodes' messages. This non-cooperative behavior aggravates overall network performance. In order to overcome lack of cooperation among nodes, it is essential to provide them with an incentive mechanism to behave in a cooperative manner and forward network packets.

In general, these methods can be classified into reputation-based and credit-based mechanisms. Reputation-based mechanisms like [3] and [4] are based on monitoring the nodes behavior from cooperation perspective and isolating misbehaving nodes. Reputation-based approach attracted some radical criticisms [6]. The first one is that it is not possible to evaluate these mechanisms in a formal manner. The second possible flaw is the constitution of a group of nodes involved in a conspiracy to maximize their utility. The third weakness in reputation-based mechanisms is their reliance upon broadcast nature of wireless networks; although considering efficient consumption of energy has made widely use of unidirectional antennas a necessity [7].

The second class of cooperation mechanisms makes use of credits in form of micro-payments [8]. Each node receives a payment for its cooperation in forwarding network messages and also pays to other nodes which participate in forwarding its messages. Hence, nodes can forward their messages using credits which have already obtained from other nodes. All of the above mentioned payments are micro-payments. The so-called credit-based mechanisms outdo reputation-based ones and recent researches have shown particular interest towards them (e.g. [4], [6], [9], [10]). From an analytical point of view, the most important aspect of these mechanisms is that by defining a utility function for a network node, it is possible to formally analyze the mechanism to determine whether nodes cooperate to forward packets using game theory.

Credit based mechanisms are confronted with the threat of nodes' dishonest behavior. In early attempts to eliminate such behavior, [4] and [10] proposed systems in which a tamper-proof hardware guarantees security issues of payments. Considering corresponding limitation, following works tried to discard the hardware [6], [9]. Sprite [6] provides an acceptable level of payments security and recent researches on cooperation problem in MANETS, has given considerable attention to it ([8], [9], [14]).

In this paper, we propose Express as a credit-based cooperation mechanism which does not rely on any tamper-proof hardware. Express makes use of several strengths of Sprite system, but simultaneously tries to both reduce computational overhead on mobile nodes and provide security against frauds of nodes. It utilizes hash chains to reduce number of digital signature operations. Every source node in Express uses digital signature only in its first transaction with any cooperative intermediate node which forwards packet(s) for it. Any following transaction with such a node requires

only hash operations instead of digital signature operations. On the other hand, Express provides incentives and fines such that rational nodes do not prefer to misbehave.

The remainder of this paper is organized as follows: Section 2 provides an overview of proposed cooperation mechanisms. In Section 3 we demonstrate different aspects of Express system. Section 4 provides formal analyses of the method using game theory. Section 5 evaluates computational overhead of Express. Finally, Section 6 concludes the paper.

II. EXPRESS

In this section, we present the overall architecture and the intuitions behind our design; the formal results will be presented in Sections 4 and 5.

A. System Architecture

The Express system is composed of Reliable Clearance Center (RCC) and several mobile nodes. Nodes are equipped with network interfaces which make it possible to transmit and receive messages in wireless network [7]. To achieve this connectivity goal, nodes make use of GPRS in outdoor areas and also can switch to 802.11 or Bluetooth in indoor areas. Contrary to network mobile nodes, the RCC does not face with limitation of energy and computation resource. To make it possible to identify the nodes belonging to a single network, it is supposed that every node has received a certificate from a certificate authority (CA). We assume that source node utilizes a source routing algorithm such as (but not restricted to) DSR [35] and for this reason is aware of the whole route to the destination.

The RCC manages credit accounts for network nodes. In a transmission path, every intermediate node gives a report of its cooperation in forwarding packets to the RCC. Although a node can save its reports in a local storage such as CompactFlash card, in order to reduce storage, each mobile node should report to the RCC whenever it switches to a fast connection and has backup power. A mobile node can also use a desktop computer as a proxy to report to the RCC.

For each packet, the source node issues a distinct warrant to each of the nodes in the packet's route. Each intermediate node would be able to make a correct packet receipt only by having this warrant and the packet's content. Instead of using digital signatures to produce each warrant, the source node takes advantage of hash chains [36].

By defining an effective incentive mechanism, the RCC provides motivation for selfish nodes to forward each other's packets. At the end of predefined time periods, the RCC clears the credits based on received reports. Source node loses credit for transmitting its packet and intermediate nodes which have participated in forwarding this packet, gain credit. Network nodes can increase their credit account either by making a deposit into their account or forwarding packet for other nodes. Connection between any mobile node and the RCC is established by a secure channel. We require that the RCC be trusted in terms of credit balance management, but the nodes do not need to trust the RCC in terms of confidentiality; because it does not receive and is not able to produce packets content from reports.

B. Operation of Nodes

This section describes operation of nodes in the proposed cooperation mechanism. In a typical packet transmission, source node S finds a path to destination node D through n intermediate nodes m_i (*for* $i=1,...,n$). D is the last intermediate node, i.e. m_n. Figure 1 depicts a typical scenario in Express. S in its first transaction with m_i produces a specific hash chain for this node and preserves it for next transactions too. Hash chain $H_i^s = (w_0^{s\to i}, w_1^{s\to i}, ..., w_{n'}^{s\to i})$ is a vector of digest-values $w_k^{s\to i}$ (*for* $k=1,...,n'$). S uses each digest-value as a warrant. The kth digest-value for ith intermediate node from S which is denoted by $w_k^{s\to i}$, is defined as follows:

$$w_k^{s\to i} = h(w_{k+1}^{s\to i}) \quad \forall k = 0,1,...,n'-1 \tag{1}$$

From collision and reversibility point of view, h is a strong hash function such as MD5 or SHA. S maintains a vector of all the hash chains $H_s = (H_1^s, ..., H_i^s, ..., H_n^s)$ that it has already produced for intermediate nodes. Although H would exhaust considerable amount of memory space with growth of n, we have utilized hash chain trees to handle this problem [13].

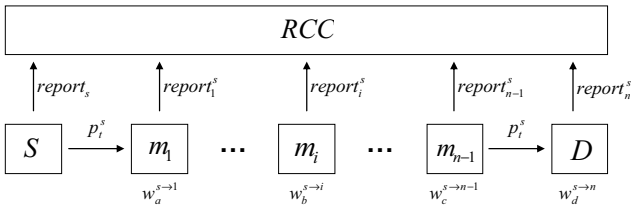

Figure 1. Mobile nodes and RCC relationships

Suppose that S wants to send its tth packet p_t^s. It adds some extra information to p_t^s through following process.

For each node m_i in the path of p_t^s:

- If it is the first time that m_i participates in forwarding a packet for S; S will generate H_i^s, create contract C_i^s and include C_i^s into p_t^s. C_i^s contains $w_0^{s\to i}$, identity of S, identity of m_i and is digitally signed by S.
- If m_i has forwarded a packet for S previously, definitely the corresponding H_i^s has been generated before. Suppose that $w_{l-1}^{s\to i}$ is the last digest-value which has been sent to m_i from S. Now S includes the pair of $(w_l^{s\to i}, l)$ into p_t^s.

S transmits the modified packet to the first intermediate node of the path. Each intermediate node m_i takes its own data (C_i^s or $(w_l^{s\to i}, l)$) from the packet and verifies it. If this is the first transaction between S and m_i, this data is C_i^s and m_i

verifies C_i^s by checking *S*'s signature on it. For next transactions, m_i verifies $w_l^{s\to i}$ by examining whether it is transformable to last previously received $w_{l-1}^{s\to i}$ or not. In this way m_i can be sure that p_t^s comes from *S*. If verification fails, m_i will drop the packet. Otherwise, m_i generates a receipt $R_i^{s,t}$ of the packet; saves this receipt and last received digest-value $w_l^{s\to i}$. Eventually m_i forwards packet for successor node in the path. This process continues until packet reaches *D*.

$R_i^{s,t}$ is the receipt that m_i generates for p_t^s and is produced as follows:

$$R_i^{s,t} = h'(p_t^s, w_l^{s\to i}) \tag{2}$$

Where, $w_l^{s\to i}$ is the digest-value that m_i receives for forwarding p_t^s. h' is a strong hash function which operates like h but takes two inputs, i.e. message's content and digest-value. h' combines its two inputs and then hashes the single result. All network nodes use an identical hash function to produce their own receipts while forwarding a packet.

To determine the length of the hash chain that it wants to produce, *S* has no need to predict the number of packets that must be sent. It creates the hash chains with an arbitrary length and based on its available memory. If number of the submitted packets did not exceed the length of the hash chain, *S* would use the rest of hash chain for its next transactions with the node that the hash chain is produced for. Hash chains are not specified for a defined transaction. Additionally, if the number packets exceeded the length of the hash chain, S would initiate a new hash chain to use it in the reminder of the transaction.

C. Mobile Nodes and the RCC

The RCC requires reports from network nodes to decide to manage credits. A report given by intermediate node m_i is as follows:

$$report_i^s = \{w_l^{s\to i}, l_i^s, C_i^s, \{R_i^{s,t}, R_i^{s,t'}, \ldots\}\} \tag{3}$$

$w_l^{s\to i}$ is the last digest-value received by m_i from *S*. l_i^s is the number of digest-values received by m_i from *S*. C_i^s is a contract signed by *S* for m_i and includes $w_0^{s\to i}$. Set $\{R_i^{s,t}, R_i^{s,t'}, \ldots\}$ includes receipts for *S*'s packets $(p_t^s, p_{t'}^s, \ldots)$ which m_i has forwarded them.

As stated before, *S* gives report to the RCC too. $report_S$ which is the report produced by *S*, is defined as follows:

$$\begin{aligned} report_s = \{&\{Path_1^s, R_a'^{s,1}, R_b'^{s,1}, \ldots\}, \ldots, \\ &\{Path_t^s, R_i'^{s,t}, R_j'^{s,t}, \ldots\}, \ldots, \\ &\{Path_{n''}^s, R_u'^{s,n''}, R_v'^{s,n''}, \ldots\}\} \end{aligned} \tag{4}$$

Set $\{Path_t^s, R_i'^{s,t}, R_j'^{s,t}, \ldots\}$ is related to packet p_t^s. In this set, $Path_t^s$ is the transmission path for p_t^s. Other elements of this set (like $R_i'^{s,t}$) are those receipts (related to p_t^s) which *S* generates for intermediate nodes of $Path_t^s$ (like node m_i).

Note that $report_i^s$ and $report_s$ are stored by nodes locally and will be forwarded as soon as a high bandwidth connection with RCC becomes available.

D. Role of the RCC

The RCC clears credit accounts periodically. As clearance decisions depend on received reports, the RCC has to make sure of their truth. Using $report_i^s$, the RCC verifies $w_l^{s\to i}$ by determining how many applications of h are required to map $w_l^{s\to i}$ into $w_0^{s\to i}$. Number of these steps should equal l_i^s. Note that $w_0^{s\to i}$ is included in C_i^s.

Observing $report_s$, the RCC should make sure that *S* has not cheated in reporting $Path_t^s$. To this end, the RCC calculates x_i^s that is the number of *S*'s packet routes in which m_i is present. x_i^s is defined as follows:

$$x_i^s = \sum_{t=1}^{n'} E(i,t) \tag{5}$$

where:

$$E(i,t) = \begin{cases} 1 & \text{if } m_i \text{ is present in } Path_t^s \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

If there is some x_i^s where $x_i^s < l_i^s$, then the RCC concludes that *S* has tried to cheat by reporting fake packet routes. The reason behind this argument is that none of the intermediate nodes are able to report more valid digest-values than what *S* has given to them. Therefore, if an intermediate node reported l_i^s digest-values from *S*, at least, it was present in routes of l_i^s packets of *S*. As a result, if the number of digest-values l_i^s that an intermediate node has reported to the RCC is greater than the intermediate node's presence which is concluded from *S*'s report (x_i^s), then surely *S* is trying to cheat the system by manipulating its reports.

The RCC tries two steps to identify which intermediate nodes have participated in forwarding p_t^s. In the first step, the RCC makes sure that an intermediate node m_i has received p_t^s. If the RCC has received $R_i^{s,t}$ and there is no difference between $R_i^{s,t}$ and $R_i'^{s,t}$, it is concluded that m_i had received p_t^s. In *Theorem 1*, using game theory it has been proved that both source and cooperative intermediate nodes always give correct receipts.

In the second step, the RCC determines how far the packet p_t^s has been forwarded. Suppose that m_k is the last node in the path that has received p_t^s. If m_k is *D* then p_t^s has reached destination. Otherwise m_k is an intermediate node which either has not forwarded p_t^s or had an unsuccessful attempt to forward it; although these two cases make no difference to the RCC. After trying above two steps, the RCC pays α units of credit to nodes $\{m_1, m_2, \ldots, m_{k-1}\}$ and β units to m_k. Value of α is greater than cost of forwarding a

packet and giving the corresponding report. Value of β is greater than cost of giving a report and $\beta < \alpha$. The RCC decreases *S*'s credit by the sum of paid credits.

E. Cheating Actions and Prevention Mechanisms

As mobile nodes of network are selfish, without a proper payment scheme, they may decide not to forward network's messages or they may try to cheat the system to increase their welfare. Particularly, a selfish node can exhibit one of the following selfish actions: first, after receiving the message, the node saves a receipt and digest-value for reporting to the RCC but does not forward the message; second, source node fakes list of paths that it reports; third, nodes fake their own receipts; fourth, the node has not received the message, but fallaciously claims that it has received the message.

Next, we will discuss why in Express mechanism nodes would not tend to do such malicious behaviors. Each mentioned attach is discussed respectively. Some of assumptions are based on theoretical results obtained in section 3.

1) Motivating nodes to forward messages

To provide incentives for selfish nodes to forward network's messages, gained credit of a node which forwards a message must be greater than that of a non-cooperative node. As illustrated before, because gained credit for the last intermediate node which has received a packet (β) is less than gained credit for previous intermediate nodes (α), every rational node prefers to participate in forwarding the packet that has been received. Equilibrium result of the packet transmission game (in *Theorem 1*) emphasizes this fact as well.

2) Motivating source node to report paths correctly

From Section 2.4 it is clear that the RCC can verifies $Path_t^s$ using $\{report_i^s; \forall i = 1,...,n\}$. If *S* cheats at reporting routes, it will be forced to pay for all of the digest-values that intermediate nodes have received from *S*. In addition, *S* will be penalized by the RCC to pay ε units of credit. ε is a small positive value. Consequently, *Theorem 2* shows that to give a correct list of its packets' routes to the RCC is a dominant strategy for *S*.

3) Motivating nodes to report correct receipts

If the RCC receives different receipts from m_i and *S*, m_i will not only gain any credits for forwarding corresponding message, but also will be fined ε units of credit. In this case, the RCC will fine the source node $\alpha + \varepsilon$ units of credit whose value is greater than cost of forwarding a packet. This ensures that source and intermediate nodes have no motivation for sending fake receipts in order to obtain more gains. Based on *Theorem 1*, if intermediate node m_i has received a message, it will report the correct receipt to the RCC, else no receipts are given. Besides, *S* always gives correct receipt for its own packets.

4) Preventing the nodes from gaining credit for messages that they have not forwarded

Section 2.4 shows that in Express a node can gain credit for forwarding a message only if it is in the path of that message and gives a correct report for it. *Theorem 2* shows that *S* has no motivation for corrupting its report of packet routes. An intermediate node is not able to report a correct receipt for a non-received packet or a packet that the node was not in its route. Firstly, a node m_i can not produce a receipt for a message of *S* so that, its receipt becomes exactly the same as *S*'s receipt, unless it has the message p_t^s and its related digest-value $w_l^{s \to i}$. Secondly, forwarding only the receipt of a message has no benefit for nodes, because each node should make its receipt based on its own digest-value (as in formula 2). The only potential threat is that intermediate node m_i generates the receipts for all of the next nodes in the path and sends these receipts to them. While this behavior has no benefit for m_i, it imposes an additional overhead of doing several extra hash operations and transmitting resulted receipts for successor nodes. Because nodes are selfish, they will not opt to do so. Hence, if a node reports a correct receipt for a packet, it surely has received the packet content and its digest-value and it has been in the packet route.

III. FORMAL ANALYSIS

Nodes of an ad hoc network select best possible strategies aiming for maximizing their own utilities. To determine best strategies for each self-interested and autonomous node, we have modeled the problem using *game theory* and obtained net result of the game by establishing its *Nash Equilibrium*. A Nash Equilibrium is a *strategy profile* having the property that no player can benefit by unilaterally deviating from its strategy.

In a typical packet transmission scenario that *S* sends a single packet to *D* through *n* intermediate nodes, there is always a possibility of transmission failure in each hop. Therefore, we suppose that with probability *P* a packet will be

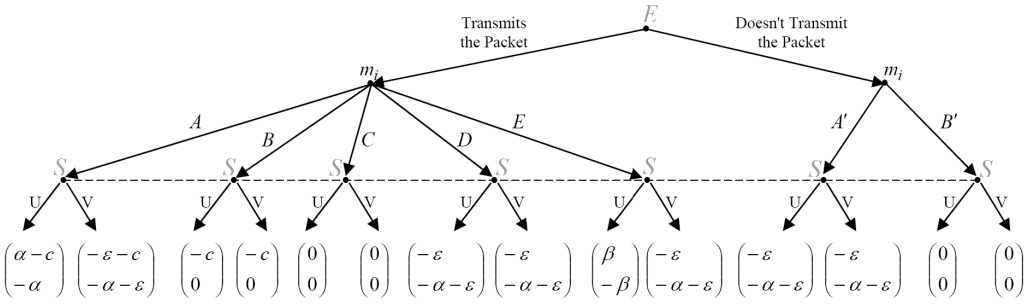

Figure 2. The extensive form of the packet transmission game. Utility of the nodes are depicted in brackets after each history. In each bracket the top value shows utility of intermediate node and bottom value shows the of source node. The dashed line indicates the knowledge set of the source node.

transmitted to an intermediate node m_i. We will show that the value of this probability has no effect in the equilibrium.

In the referred packet transmission scenario, m_i has different available strategies depending on whether or not it has received the packet. If m_i receives the packet it can follow one of these five strategies:

- A : It forwards the packet and reports a correct receipt (to ensure that it has received the packet correctly);
- B : It forwards the packet, but does not report any receipt for it;
- C : It does not forward the packet and does not report any receipt for it;
- D : It does not forward the packet, but reports a manipulated receipt;
- E : It does not forward the packet and reports a correct receipt (to ensure that it has received the packet correctly);

If m_i doesn't receive the packet, it has these two strategies on hand:

- A' : It reports a manipulated receipt (to deceive the RCC into showing that it has received the packet);
- B' : It does not report any receipt (to denote that it has not received the packet).

On the other hand S should select its own strategy using incomplete information. It does not know whether its packet is received by m_i. It has also no idea what strategy m_i has chosen. Whatever the situation is, S can follow one of these two strategies:

- U : It reports a correct receipt for its packet;
- V : It reports a manipulated receipt of its packet to make its receipt dissimilar to m_i's one in the hope of evading payment to m_i (Note that the RCC pays to m_i only if both receipts from m_i and S are exactly the same).

The *extensive form* of the game is depicted in Figure 2. There are three players in the game: *Source node* (S), *Intermediate node* (m_i) and *Environment* (E) whose duty is to transmit the S's packet to m_i. As shown in the figure, E plays first and it selects its left action with probability P and its right action with probability *1-P*. m_i plays in the next turn and S plays last. Based on the history that has happened in the game, utility of nodes would be different. A history is a sequence of strategies that are selected by players. Therefore, every history is a distinct path in the game's tree from the root to a leaf. At the end of each history in the figure, utility gained by each player is shown inside brackets. In each bracket the top value is utility of m_i and the bottom value is utility of S. As stated before, α is the credit that an intermediate node gains for forwarding a packet, β is the credit that is given to the last node of the packet's transmission path which has reported a correct receipt for the packet, ε is the fine that the RCC imposes on nodes if they report dissimilar receipts and finally, c is the cost of forwarding a packet to the next node. The dashed-line indicates the *information set* of S.

TABLE I. STRATEGIC FORM OF THE PACKET TRANSMISSION GAME.

	U	V
A, A'	$\begin{pmatrix} P\times(\alpha-c)+(1-P)\times(-\varepsilon) \\ P\times(-\alpha)+(1-P)\times(-\alpha-\varepsilon)\end{pmatrix}$	$\begin{pmatrix} P\times(-\varepsilon-c)+(1-P)\times(-\varepsilon) \\ -\alpha-\varepsilon\end{pmatrix}$
A, B'	$\begin{pmatrix} P\times(\alpha-c) \\ P\times(-\alpha)\end{pmatrix}$	$\begin{pmatrix} P\times(-\varepsilon-c) \\ P\times(-\alpha-\varepsilon)\end{pmatrix}$
B, A'	$\begin{pmatrix} P\times(-c)+(1-P)\times(-\varepsilon) \\ (1-P)\times(-\alpha-\varepsilon)\end{pmatrix}$	$\begin{pmatrix} P\times(-c)+(1-P)\times(-\varepsilon) \\ (1-P)\times(-\alpha-\varepsilon)\end{pmatrix}$
B, B'	$\begin{pmatrix} P\times(-c) \\ 0\end{pmatrix}$	$\begin{pmatrix} P\times(-c) \\ 0\end{pmatrix}$
C, A'	$\begin{pmatrix} (1-P)\times(-\varepsilon) \\ (1-P)\times(-\alpha-\varepsilon)\end{pmatrix}$	$\begin{pmatrix} (1-P)\times(-\varepsilon) \\ (1-P)\times(-\alpha-\varepsilon)\end{pmatrix}$
C, B'	$\begin{pmatrix} 0 \\ 0\end{pmatrix}$	$\begin{pmatrix} 0 \\ 0\end{pmatrix}$
D, A'	$\begin{pmatrix} -\varepsilon \\ -\alpha-\varepsilon\end{pmatrix}$	$\begin{pmatrix} -\varepsilon \\ -\alpha-\varepsilon\end{pmatrix}$
D, B'	$\begin{pmatrix} P\times(-\varepsilon) \\ P\times(-\alpha-\varepsilon)\end{pmatrix}$	$\begin{pmatrix} P\times(-\varepsilon) \\ P\times(-\alpha-\varepsilon)\end{pmatrix}$
E, A'	$\begin{pmatrix} P\times(\beta)+(1-P)\times(-\varepsilon) \\ P\times(-\beta)+(1-P)\times(-\alpha-\varepsilon)\end{pmatrix}$	$\begin{pmatrix} -\varepsilon \\ -\alpha-\varepsilon\end{pmatrix}$
E, B'	$\begin{pmatrix} P\times(\beta) \\ P\times(-\beta)\end{pmatrix}$	$\begin{pmatrix} P\times(-\varepsilon) \\ P\times(-\alpha-\varepsilon)\end{pmatrix}$

To establish the game's equilibrium, we have transformed the extensive form of the game into a strategic form one. The strategic form of the game is shown in Table 1. Columns of the table represent possible strategies for S (i.e. U and V). Each row of the table is a possible pair strategy of m_i. Cells of the table contain utility pairs that players gain probably in each strategy profile. Satisfying conditions: $\alpha, \beta, \varepsilon, c > 0$ and $\alpha > \beta + c$, the strategy profile $((A, B'), U)$ is the game's exclusive equilibrium. Hence, if m_i receives a packet from S it will select strategy A and if it does not receive the packet it will select strategy B'. S will always select the strategy U.

Theorem 1: Satisfying conditions: $\alpha, \beta, \varepsilon, c > 0$ and $\alpha > \beta + c$, strategy profile $((A, B'), U)$ is the equilibrium of packet transmission game. It means that if m_i receives the packet of S from previous node in the transmission path, it will report a correct receipt for it. If m_i does not receive the message of S, it will not report any receipt for that message. In addition, S will always report a correct receipt for its messages.

Theorem 2: S will not try to cheat the RCC by changing its sent packet's route in its report.

Proof: According to description of the Express, if the RCC finds out that S has cheated in reporting its packet's route, the RCC will pay to the intermediate nodes for all of the digest-values that they have reported from S and the RCC will decrease the whole credit from S. Being a cheat, S loses the following credit:

$$c_f^s = \left[\sum_{i=1}^{n} (l_i^s \times \alpha) \right] + \varepsilon \quad (7)$$

where l_i^s is number of digest-values that m_i has received from *S*, α is the credit that each node gains for forwarding the message and ε is the penalty that *S* should pay. However, if message routes were correctly reported to the RCC, by checking the receipts, the RCC could determine for which one of digest-values that a node had reported, it had also forwarded the message. Then, it transfers the credits from *S*'s account to other nodes only for digest-values that they forwarded its corresponding message, not for all of the digest-values. In addition, when a packet has not reached its destination, the last node that reported the receipt will gain β instead of α, which decreases the amount of credit that *S* will lose. Therefore, by being honest, the amount of credit that *S* will lose is c_h^s; defined as follows:

$$c_h^s \le \sum_{i=1}^{n} (k_i^s \times \alpha) \quad (8)$$

where k_i^s is the number of messages that m_i has forwarded for *S* and $k_i^s \le l_i^s$.

Utilities of *S* for two strategies *f* (which stands for fraud) and *h* (which stands for honest) are $u(f) = -c_f^s$ and $u(h) = -c_h^s$. It is clear that $u(h) > u(f)$. Consequently, reporting the correct list of its packet routes would be a *dominant strategy* for *S*.

IV. Conclusion

Mobile ad-hoc networks exhibit new vulnerabilities to malicious attackers and denial of cooperation. In this paper, we proposed a cooperation mechanism in which security issues and providing rational incentives for cooperation is granted. We believe that a reliable solution should be both secure and efficient. All sorts of known cheats are prevented by Express. Moreover, additional processes imposed by Express are significantly lighter than other works we have reviewed. The main idea to reduce processes is to use hash chains instead of digital signatures. We studied the behavior of selfish nodes in our proposed mechanism analytically.

References

[1] J. Schiller, Mobile Communications, 1st ed., Addison-Wesley Professional, 2000.

[2] E. Huang, J. Crowcraft, and I. Wassell, "Rethinking Incentives for Mobile Ad Hoc Networks," in Proc. SIGCOMM'04 Workshops, pp. 191-196. 2004,

[3] S. Buchegger and J.Y. Le Boudec, "Performance analysis of the CONFIDANT protocol (Cooperation of nodes—Fairness in dynamic ad hoc networks)," in Proc. IEEE/ACM MobiHOC, 2002.

[4] L. Buttyán and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," ACM/Kluwer Mobile Networks and Applications (MONET), vol. 8, no. 5, pp. 579–592, Oct. 2003.

[5] S. Zhong, J. Chen, and Y. R.Yang, "Sprite:Asimple, cheat-proof, creditbased system for mobile ad hoc networks," in Proc. INFOCOM, 2003.

[6] M. Stemm and R. H. Katz, "Vertical handoffs in wireless overlay networks," Mobile Networks and Applications, vol. 3, no. 4, pp. 335–350, 1998.

[7] M. Jakobsson, J.-P.Hubaux and L. Buttyan "A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks." Ser. Lecture Notes in Computer Science. Berlin / Heidelberg, Germany, 2004, vol. 2742. pp. 15-33.

[8] P. Marbach and Y. Qiu, "Cooperation in wireless ad hoc networks: a market-based approach," IEEE/ACM Transactions on Networking (TON), vol. 13, issue 6, pp. 1325-1338 , 2005.

[9] L. Bla˘zevic´, L. Buttyán, S. ˘ Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, "Self-organization in mobile ad hoc networks: The approach of terminodes," IEEE Commun. Mag., vol. 39, no. 6, pp. 166–174, Jun. 2001.

[10] M. Hauspie and I. Simplot-Ryl, "Cooperation in ad hoc networks: Enhancing the virtual currency based models," In Proc. the 1st ACM International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense), 2006.

[11] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," In Proc. SIGCOMM, 1996.

[12] R. L. Rivest and A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes,", Lecture Notes in Computer Science, volume 1189 pp. 69–87. Springer, 1997.

[13] S. Yen, L. Ho and C. Huang, "Internet Micropayment Based on nbalanced One-way Binary Tree," In Proc. of CrypTEC '99, pp.155-62, 1999.