Volume: 2 Issue: Issue- III Month of publication: November 2014

DOI:

www.ijraset.com
SJ Impact Factor-3.995

Special Issue-3, November 2014
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Efficient and Accurate Misbehavior Detection Scheme in Adversary Environment

Kanagarohini.V[1], Ramya.K[2]

[1]Student, [2]Guide, Sree Sowdambika College of Engineering

Abstract— ***Misbehaviour detection is regarded as a great challenge in the adversary environment because of distinct network characteristics. Harmful and egocentric behaviours illustrate an insecure threat against routing in delay tolerant networks (DTNs). In order to address this, in this paper, we propose iTrust, a probabilistic misbehaviour detection scheme for efficient and accurate misbehaviour detection in DTNs. Our iTrust scheme introduces the periodically available Trusted Authority (TA) to estimate the node's behavior based on the collected routing evidences. To further enhance the power of the proposed model, we associate the detection probability with node's reputation for effective inspection.***

Keywords—Delay Tolerant Networks, Trusted Authority, Reputation.

I. INTRODUCTION

Delay Tolerant Network is a communication network designed to tolerate long delays and outages. The current networking technology depends on a set of basic assumptions that are not true in all environments. The first and most important assumption is that an end-to-end connection exists from the source to the destination. This assumption can be easily contravened due to mobility, power saving etc. Examples of such networks are sensor networks with scheduled infrequent connectivity, vehicular DTNs that publish local ads, traffic reports, parking information [1]. Delay tolerant network (DTN) is an attempt to extend the reach of networks. It give an assurance to enable communication between "challenged" networks.

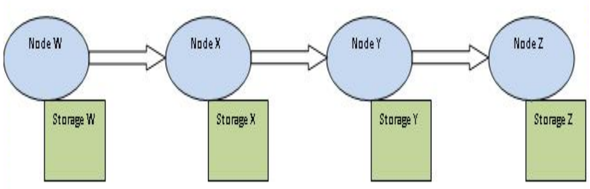


Fig. 1, Delay Tolerant Networking Environment

Delay Tolerant Networks have unique characteristics like lack of contemporaneous path, short range contact high variation in network conditions, difficult to predict mobility patterns and long feedback delay. Because of these unique characteristics the Delay Tolerant Networks (DTNs) move to an approach known as "store-carry-and-forward" strategy where the bundles can be sent over the existing link and buffered at the next hop until the next link in the path appears and the routing is determined in an "opportunistic" fashion. In DTNs a node could misbehave by refusing to forward the packets, dropping the packets even when it has the potential to forward (e.g., sufficient memory and meeting opportunities) or modifying the packets to launch attacks. These types of malicious behaviors are caused by rational or malicious nodes, which try to maximize their own benefits. Such malicious activities pose a serious threat against network performance and routing. Hence a trust model is highly enviable for misbehavior detection and attack mitigation.

II. RELATED WORK

Routing misbehavior detection and mitigation has been well crammed in traditional mobile ad hoc networks. These methodologies use neighborhood monitoring or destination acknowledgement (ACK) to detect dropping of packets [2]. In the mobile ad hoc networks (MANET) first complete route is established from source to destination, before transmitting the packet. But in DTN the nodes are intermittently connected, hence there is no possibility for route discover and it has other unique characteristics like dynamic topology, short range contact, long feedback delay which made the neighborhood monitoring unsuitable for DTN. Although many routing algorithms [3, 4, 5, 6, 7] have been proposed for DTNs, most of them do not consider the node's willingness to forward the packet and implicitly assume that a node is willing to forward packets for all others. They may not work well since some packets are forwarded to nodes unwilling to relay, and will be dropped.

There are quite a few proposals for misbehaviour detection which are based on forward history verification (e.g., multi layer formation [8]) and by providing encounter tickets [9], which incur high transmission overhead as well as high verification cost. Different from the exiting works in which the Trusted Authority (TA) performs the auditing based on checking the contact history [10], is critical and time consuming. Our proposed system uses nectar protocol for selecting the appropriate intermediate node such that the inspection or auditing process can be simplified and the packet dropping rate can be considerably reduced. To achieve a tradeoff between detection cost and security, our Trust model relies on inspection game [11] based on game theory. This introduces a periodically available Trusted Authority (TA) to judge the nodes based on collected routing evidences. Our Trust model jointly considers the incentive and malicious node detection scheme in the single framework along with the effective nectar protocol for selecting the appropriate intermediate node. The contributions of this paper can be summarized as follows.

1. We propose a general misbehavior detection framework based on a series of newly introduced data forwarding evidences. The proposed evidence framework could not only detect various misbehaviors but also be compatible to various routing protocols.

2. Malicious node detection is carried out by the Trusted Authority (TA) based on the evidences generated by nodes, which are selected by the application of protocol.

3. Hence packet dropping rate can be considerably reduced and the performance of the network can be improved.

III. PRELIMINARY

A. System Model

Delay Tolerant Network consists of mobile devices owned by individual users. Each node i is assumed to have a unique ID Ni and a corresponding public/private key pair. We assume that each node must pay a deposit before it joins the network, and the deposit will be paid back after the node leaves if there is no misbehavior activity of the node. we assume that a periodically available TA exists so that it could take the responsibility of misbehavior detection in DTN. For a specific detection target Ni, TA will request Ni's forwarding history in the global network. Therefore, each node will submit its collected Ni's forwarding history to TA via two possible approaches.

In a pure peer-to-peer DTN, the forwarding history could be sent to some special network components (e.g., roadside unit (RSU) in vehicular DTNs or judge nodes in [10]) via DTN transmission. In some hybrid DTN network environment, the transmission between TA and each node could be also performed in a direct transmission manner (e.g., WIMAX or cellular networks [14]). We argue that because the misbehavior detection is performed Periodically, the message transmission could be performed in a batch model, which could further reduce the transmission overhead.

B. Routing Model

We adopt the single-copy routing mechanism such as First Contact routing protocol, and we assume the communication range of a mobile node is finite. Thus, a data sender out of destination node's communication range can only transmit packetized data via a sequence of intermediate nodes in a multihop manner. Our misbehaving detection scheme can be applied to delegation-based routing protocols or multicopy-based routing ones, such as MaxProp [18] and ProPHET [19]. We assume that the network is loosely synchronized (i.e., any two nodes should be in the same time slot at any time).

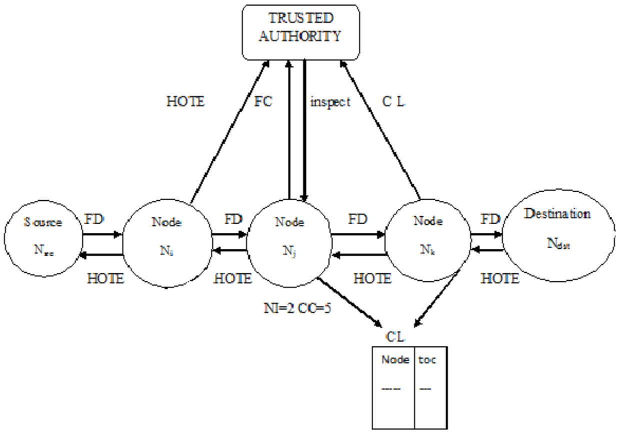


Fig 2: Trust model architecture

HOTE – Hand Over Task Evidence
FD – Forwarding
FC – Forward Chronicle

CL- Contact Log
NI - Neighborhood Index
Ni, Nj, Nk – Intermediate Nodes
CC – Contact Counter
TOC- time of contact

C. Adversary Model

First of all, we assume that each node in the networks is rational and a rational node's goal is to maximize its own profit. In this work, we mainly consider two kinds of DTN nodes: selfish nodes and malicious nodes. Due to the selfish nature and energy consuming, selfish nodes are not willing to forward bundles for others without sufficient reward. As an adversary, the malicious nodes arbitrarily drop others' bundles (black hole or gray hole attack), which often take place beyond others' observation in a sparse DTN, leading to serious performance degradation. Note that any of the selfish actions above can be further complicated by the collusion of two or more nodes.

IV. ROUTING TESTIMONY GENERATION PHASE

The basic iTrust has two phases, including routing testimony generation phase and auditing phase. In the evidence generation phase, the nodes will generate contact and data forwarding evidence for each contact or data forwarding. In the subsequent auditing phase, TA will distinguish the normal nodes from the misbehaving nodes. For an example, we take a three-step data forwarding process. consider node X has packets, which will be delivered to node Z. If node X meets another node Y that could helps to forward the packets to Z, X will replicate and forward the packets to Y. Afterwards, Y will forward the packets to Z when Z arrives at the transmission range of Y. In this process, we define three kinds of data forwarding evidences that could be used to judge if a node is a malicious one or not:

A. Hand Over Task Evidence

Hand over Task evidences are used to record the number of routing tasks assigned from the upstream nodes to the target node *Nj.* We assume that source node (Nsrc) has message M, in order to forward to the destination (Ndst). For simplicity of presentation, consider that message is stored at the intermediate node (Ni), when Nj comes within the transmission or radio range of Ni ,then it will determine by means of nectar protocol whether to choose node j(Nj) as the intermediate node or not, in order to forward message M to the destination.

If node j (Nj) is the chosen next node then the flag bit will be enabled (or *flag* = 1) and the Task evidence $E^{i\rightarrow j}_{task}$ need to be generated, to demonstrate that a new task has been assigned from node i (Ni) to node j (Nj), Where Tts and TExp refer to the time stamp and the expiration time of the packets. we set $M^{i>j}_{M=}${M,Nsrc, flag, Ni, Nj,Ndst, Tts, TExp, Sigsrc}, where Sigsrc= Sigsrc(H(M,Nsrc, Ndst, TExp)) refers to the signature generated by the source nodes on message M. Node Ni generates the signature Sigi =SIGi{ $M^{i->j}_{M}$ } to indicate that this forwarding task has been delegated to node Nj while node Nj generates the signature Sigj =SIGj{ $IM^{i->j}_{M}$ } to show that Nj has accepted this task. Therefore, we obtain the delegation task evidence as follows:

$$E^{i\rightarrow j}_{task} = \{M^{i->j}_{M}, Sig_i, Sig_j\} \quad (1)$$

B. Forwarding Chronicle evidence

When Nj meets the next intermediate node Nk, Nj will check if Nk is the desirable next intermediate node in terms of a specific routing protocol. If yes Nj will forward the packets to Nk, who will generate a forwarding history evidence to demonstrate that Nj has successfully finished the forwarding task. Nk will generate a signature Sig_k=SIG_k{H($M^{j->k}_{M)}$} to demonstrate the authenticity of forwarding history evidence. Therefore, the complete forwarding history evidence is generated by Nk as follows:

$$E^{j->k}_{forward} = \{M^{j->k}_{M}, Sig_k\} \quad (2)$$

In the audit phase, the node which is inspected will submit its forwarding history evidence to TA to demonstrate that it has tried its best to accomplish the routing tasks, which are defined by hand over task evidences.

C. Contact log evidence

Whenever two nodes meet, a new contact log is generated and the neighbourhood index is updated accordingly. Each node also maintains a contact counter, which keeps track of how often the nodes meet each other. When two nodes Nj and Nk meet, a new contact log $E^{j<->k}_{contact}$ will be generated. Suppose that $M^{j<->k}$ = {N_j ,N_k, Tts}. N_j and N_k will generate their signatures Sig_j = SIG_j {H ($M^{j->k}$)} and Sig_k = SIG_k{H($M^{j<->k}$)}. Therefore, the contact history evidence could be obtained as follows:

$E^{j<->k}_{contact} = \{M^{j<->k}, Sig_j, Sig_k\}$ (3)

The contact log will be stored at both of meeting nodes. In the audit phase both the nodes will submit their logs to the TA. Maintenance of contact history could prevent the black hole or grey hole attack. The nodes chosen by the nectar protocol with sufficient contact with other users, but if it fails to forward the data, will be regarded as a malicious or selfish one.

V. AUDITING PHASE

Since the selection of intermediate node is based on the Nectar protocol, the dropping rate of packect is reduced considerably. In order to further improve the network performance and to avoid packet dropping, our trust model introduces the Trusted Authority (TA), which periodically launches the investigation request.

In the auditing phase, the Trusted Authority (TA) will send the investigation request to node Nj in a global network during a certain period [t1, t2]. Then, given N as the set of nodes in the network, each node in the DTN will submit it's collected $\{E^{i->j}_{task}, E^{j->k}_{forward}, E^{j<->k}_{contact}\}$ to TA. After collecting all of the evidences related to Nj , TA obtains the set of task evidence S_{task}, the set of messages forwarded $S_{forward}$ and the set of contacted nodes $S_{contact}$. To check if a suspected node Nj is malicious or not, TA should check if any message forwarding request has been honestly fulfilled by Nj.

A. Reliable data forwarding with adequate users

A normal user will honestly follow the routing protocol by forwarding the messages to the sufficient users. Therefore, the given message m is in Stask, the data is forwarded to the presence of adequate users. The requested message has been forwarded to the next hop, the chosen next hop nodes are desirable nodes according to a specific DTN routing protocol, and the number of forwarding copies satisfy the requirement defined by a multicopy forwarding routing protocol.

B. Reliable data forwarding with inadequate users

A normal user will also honestly perform the routing protocol but fail to achieve the desirable results due to lack of adequate users. Therefore, given the message m is in Stask, the data is forwarded to the presence of adequate users. There are two cases are here. First case is that there is no contact during period [Tts (m), t2]. The second case is that only a limited number of contacts are available in this period and the number of contacts is less than the number of copies required by the routing protocols. In both cases, even though the DTN node honestly performs the routing protocol, it cannot fulfill the routing task due to lack of sufficient contact chances. We still consider this kind of users as honest users.

C. A misbehaving data forwarding with/without adequate users

A misbehaving node will drop then packets or refuse to forward the data even when there are sufficient contacts. There are three cases are here. The first case is the forwarder refuses to forward the data even when the forwarding opportunity is available. The second case is that the forwarder has forwarded the data but failed to follow the routing protocol. The last case is that the forwarder agrees to forward the data but fails to propagate the enough number of copies predefined by a multicopy routing protocol

VI. ALGORITHM FOR MISBEHAVIOR DETECTION

The TA judges if node Nj (Suspected node) is malicious or not by triggering the Malicious node detection algorithm. Where node j is the suspected malicious node, Stask is the set of hand over task evidence, $S_{forward}$ is the set of forward chronicle, and R is the set of contacted nodes, Nk(m) as the set of next-hop nodes chosen for message forwarding, C represents the punishment (lose of deposit), we denotes the compensation (virtual currency or credit) paid by TA.

Algorithm 1. The basic misbehavior detection algorithm

```
procedure BASICDETECTION ((j,$S_{task}$, $S_{forward}$, [t1, t2], R))
    for each m is in $S_{task}$ do

      if m is not in $S_{forward}$ and $R\neq0$ then
          return 1
      else if m is in $S_{forward}$ and $N_k$(m$_{)}$ is not in R then
          return 1
    else if m is in $S_{forward}$ and $N_k(m_)$ is in R then
  |Nk(m)| is less than D then
      return 1
    end if
  end for
  return 0
end procedure
```

In this algorithm, we introduce Basic Detection, which takes j, S_{task}, $_{Sforward}$, [t1, t2], R, D as well as the routing requirements of a specific routing protocol R, D as the input, and output the detection result “1” to indicate that the target node is a misbehavior or “0” to indicate that it is an honest node. To prevent malicious users from providing fake delegation/forwarding/contact evidences, TA should check the authenticity of each evidence by verifying the corresponding signatures, which introduce a high transmission and signature verification overhead.

Algorithm 2. The Proposed Malicious Node Detection algorithm

```
initialize the number of nodes n
for i←1 to n do
  generate a random number mi from 0 to 10n _ 1
if mi/$10^n$ < pb then
ask all the nodes (including node i) to provide evidence about node i
if Basic Detection(I,Stask,Sforward,[t1, t2],R,D) then
give a punishment C to node i
Else
pay node i the compensation w
end if
Else
pay node i the compensation w
end if
end for
```

The above algorithm shows the details of the proposed probabilistic misbehavior detection scheme. For a particular node i, TA will launch an investigation at the probability of pb. If i could pass the investigation by providing the corresponding evidences, TA will pay node i a compensation w; otherwise, i will receive a punishment (lose its deposit).

VII. PROBABILITY FIXING INSPIRED BY GAME THEORY

There are two strategies available for the trusted authority and the nodes. The Trusted Authority can choose inspecting (I) or not inspecting (N). Each node also has two strategies, forwarding(F) and offending (O).

Theorem:

If TA inspects at the probability of Pb = g+£/w+C in Trust Model, a rational node must choose forwarding strategy, and the TA will get a higher profit than it checks all the nodes in the same round.

Proof:

This is a static game of complete information, though no dominating strategy exists in this game, there is a mixed Nash Equilibrium point.

If the node chooses offending strategy, its payoff is

$$\pi_w(S) = -C \cdot (g + £/w + C) + w \cdot (g + £/w + C) = w - g - £$$

If the node chooses forwarding strategy, its payoff is

$$\pi_w(W) = Pb \cdot (w - g) + (1 - Pb) \cdot (w - g) = w - g$$

The latter one is obviously larger than the previous one. Therefore, if TA chooses the checking probability $g+\_/w+C$, a rational node must choose the forwarding strategy. Furthermore, if TA announces it will inspect at the probability $Pb = g+£/w+C$ to every node, then its profit will be higher than it checks all the nodes, for

$$v - w - (g + £ /w + C) \cdot h > v- w- h$$

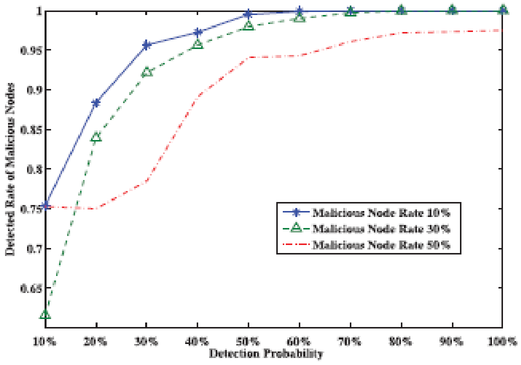
Here the latter part in the inequality is the profit of TA when it checks all the nodes. Note that the probability that a malicious node cannot be detected after *k* rounds is (1− *g*+£ /*w*+*C*)*k* → 0, if *k*→∞. Thus it is almost impossible that a malicious node cannot be detected after a certain number of rounds.

VIII. EXPERIMENT RESULTS

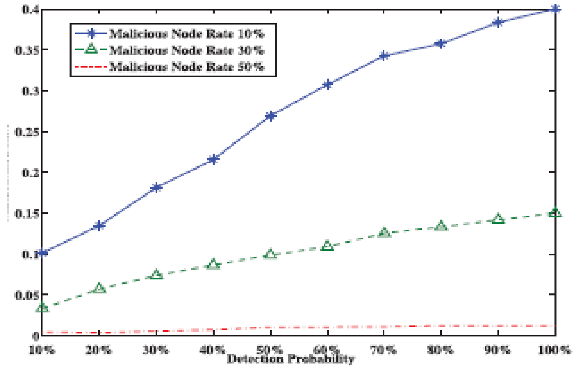
We set up the experiment environment with the opportunistic networking environment (NS2) simulator, which is designed for evaluating DTN routing and application protocols. In our experiment, we adopt the First Contact routing protocol, which is a single-copy routing mechanism. We set the time interval T to be about 3 hours as the default value, and we deploy 50, 80, 100 nodes on the map, respectively. With each parameter setting, we conduct the experiment for 100 rounds. We use the packet loss rate (PLR) to indicate the misbehavior level of a malicious node. In DTNs, when a node's buffer is full, a new received bundle will be dropped by the node, and PLR denotes the rate between the dropped bundles out of the received bundles. But, a malicious node could pretend no available buffer and, thus, drop the bundles received. Thus, PLR actually represents the misbehavior level of a node. For example, if a node's PLR is 1, it is totally a malicious node who launches a black hole attack. If a node's PLR is 0, we take it as a normal node. Further, if 0 < PLR < 1, the node could launch a gray hole attack by selectively dropping the packets. In our experiment, we use the detected rate of the malicious nodes to measure the effectiveness of iTrust, and we take all the nodes whose PLR larger than 0 as the malicious ones. On the other hand, since a normal node may also be identified as the malicious one due to the depletion of its buffer, we need to measure the false alert of iTrust and show that iTrust has little impact on the normal users who adhere to the security protocols. Thus, we use the misidentified rate to measure the false negative rate. Moreover, we evaluate the transmission overhead $\text{Cost}_{\text{transmission}}$ and verification overhead $\text{Cost}_{\text{verification}}$ in terms of the number of evidence transmission and verification for misbehavior detection. In the next section, we will evaluate the effectiveness of iTrust under different parameter settings.

A. The Impact of Percentage of Malicious Nodes on iTrust

We use malicious node rate (MNR) to denote the percentage of the malicious nodes of all the nodes. In this experiment, we consider the scenarios of varying MNR from 10 to 50 percent. In this experiment, PLR is set to be 1, and the velocity of 80 nodes varies from 10:5 to 11:5 m=s. The message generation time interval varies from 25 to 35 s, and the TTL of each message is 300 s. The experiment result is shown in Fig. 3. Fig. 3a shows that three curves have the similar trends, which indicate that iTrust could achieve a stable performance with different MNRs. Even though the performance of iTrust under high MNR is lower than that with low MNR, the detected rate is still higher than 70 percent. Furthermore, the performance of iTrust will not increase a lot when the detection probability exceeds 20 percent, but it is good enough when the detection probability is more than 10 percent. Thus, the malicious node rate has little effect on the detected rate of malicious nodes. That means iTrust will be effective, no matter how many malicious nodes there are. Further, a high malicious node rate will help reduce the misidentified rate as shown in Fig. 3b because the increase of the malicious nodes will reduce the proportion of the normal nodes who will be misidentified.



(a) Detected rate of malicious nodes with different MNRs

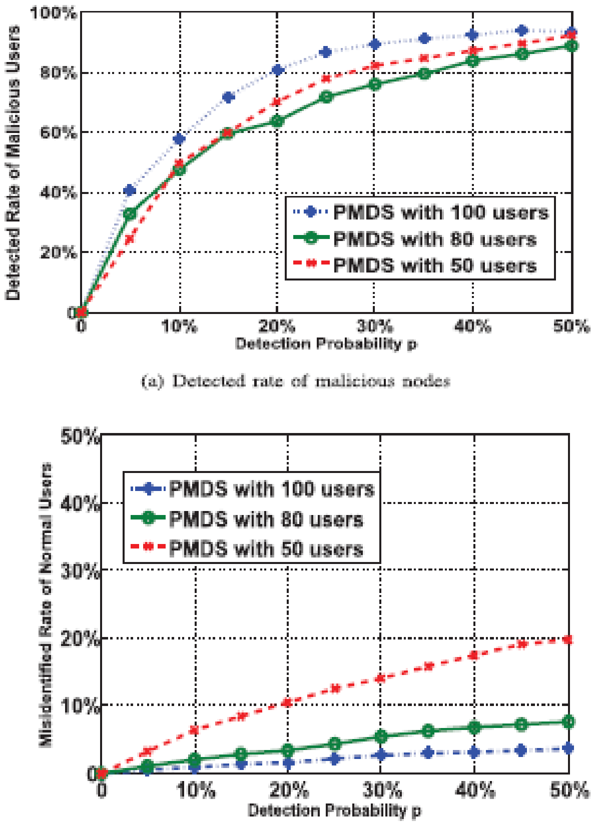


(b) Misidentified rate with different MNRs

Fig. 3. Experiment results with different MNRs.

B. The Evaluation of the Scalability of iTrust

First, we evaluate the scalability of iTrust, which is shown in Fig. 4. As we predict in (12), the number of nodes will affect the number of generated contact histories in a particular time interval. So we just measure the detected rate (or successful rate) and misidentified rate (or false positive rate) in Fig. 4. Fig. 4a shows that when detection probability p is larger than 40 percent, iTrust could detect all the malicious nodes, where the successful detection rate of malicious nodes is pretty high. It implies that iTrust could assure the security of the DTN in our experiment. Furthermore, the misidentified rate of normal users is lower than 10 percent when user number is large enough, as shown in Fig. 4b, which means that iTrust has little impact on the performance of DTN users. Therefore, iTrust achieves a good scalability.



(a) Detected rate of malicious nodes

(b) false rate of misidentified nodes

Fig. 4. Experiment results with user number of 100, 80, 50.

C. *The Impact of Various Packet Loss Rate onITrust*

In the previous section, we have shown that iTrust could also thwart the gray hole attack. In this section, we evaluate the performance of iTrust with different PLRs. In this experiment, we measure the scenarios of varying PLR from 100 to 80 percent. We set MNR as 10 percent, and the speed of 80 nodes varying from 10:5 to 11:5 m=s. The message generation interval varies from 25 to 35 s, and the TTL of each message is 300 s. The experiment result and PLRs have little effect on the performance of iTrust, as shown in Fig. 5. This implies iTrust will be effective for both black hole attack and gray hole attack. The misidentified rate is not affected by PLRs either. It is under 8 percent when the detection probability is under 10 percent. Thus, the variation of PLR will not affect the performance of iTrust.

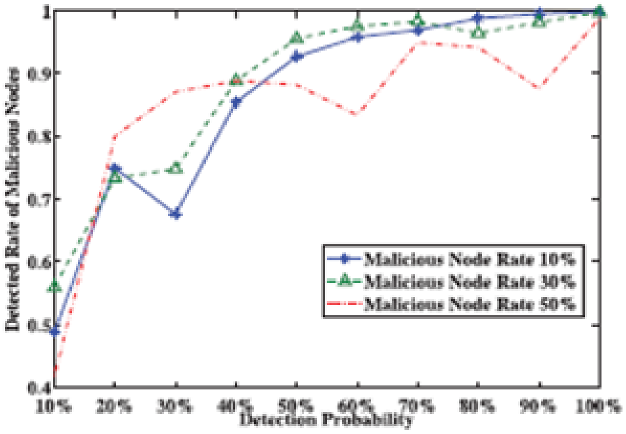


Fig. 5. Experiment results with different PLRs.

IX. CONCLUSION

In this paper we propose a Trust Model which could effectively detect the malicious node and ensures secure transmission of data. The selection of neighbour node is based on AODV protocol, by which the packet dropping rate is considerably reduced and it also simplifies the work of Trusted Authority (TA). We also reduce the detection overhead by introducing the Trusted Authority (TA) designed on the basis of inspection theory, in a periodic fashion.

REFERENCES

[1] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A New VANETBased Smart Parking Scheme for Large Parking Lots," Proc. IEEE INFOCOM '09, Apr. 2009.

[2] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know the Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing," Proc. IEEE INFOCOM '10, 2010.

[3] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay-Tolerant Networks," Proc. IEEE INFOCOM '10, 2010.

[4] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks," IEEE Trans. Vehicular Technology, vol. 58, no. 8, pp. 828-836, 2009.

[5] H. Zhu, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "SLAB: Secure Localized Authentication and Billing Scheme for Wireless Mesh Networks," IEEE Trans. Wireless Comm., vol. 17, no. 10, pp. 3858- 3868, Oct. 2008.

[6] Q. Li and G. Cao, "Mitigating Routing Misbehavior in Disruption Tolerant Networks," IEEE Trans. Information Forensics and Security, vol. 7, no. 2, pp. 664-675, Apr. 2012.

[7] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proc. ACM MobiCom '00, 2000.

[8] R. Lu, X. Lin, H. Zhu, and X. Shen, "Pi: A Practical Incentive Protocol for Delay Tolerant Networks," IEEE Trans. Wireless Comm., vol. 9, no. 4, pp. 1483-1493, Apr. 2010.

[9] F. Li, A. Srinivasan, and J. Wu, "Thwarting Blackhole Attacks in Disruption-Tolerant Networks Using Encounter Tickets," Proc. IEEE INFOCOM '09, 2009.

[10] E. Ayday, H. Lee, and F. Fekri, "Trust Management and Adversary Detection for Delay-Tolerant Networks," Proc. Military Comm. Conf. (Milcom '10), 2010.

[11] D. Fudenberg and J. Tirole, Game Theory. MIT Press, 1991.

[12] M. Rayay, M.H. Manshaeiy, M. Flegyhziz, and J. Hubauxy, "Revocation Games in Ephemeral Networks," Proc. 15th ACM Conf. Computer and Comm. Security (CCS '08), 2008.

[13] S. Reidt, M. Srivatsa, and S. Balfe, "The Fable of the Bees: Incentivizing Robust Revocation Decision Making in Ad Hoc Networks," Proc. 16th ACM Conf. Computer and Comm. Security
(CCS '09), 2009.

[14] B.B. Chen and M.C. Chan, "Mobicent: A Credit-Based Incentive System for Disruption-Tolerant Network," Proc. IEEE INFOCOM '10, 2010.

[15] S. Zhong, J. Chen, and Y.R. Yang, "Sprite: A Simple Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," Proc. IEEE INFOCOM '03, 2003.

[16] J. Douceur, “The Sybil Attack,” Proc. Revised Papers from the First Int’l Workshop Peer-to-Peer Systems (IPTPS ’01), 2001.

[17] R. Pradiptyo, “Does Punishment Matter? A Refinement of the Inspection Game,” Rev. Law and Economics, vol. 3, no. 2, pp. 197- 219, 2007.

[18] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, “Maxprop: Routing for Vehicle-Based Disruption-Tolerant Networks,” Proc. IEEE INFOCOM ’06, 2006.

[19] A. Lindgren and A. Doria, “Probabilistic Routing Protocol for Intermittently Connected Networks,” draft-lindgren-dtnrg-prophet- 03, 2007.

[20] W. Gao and G. Cao, “User-Centric Data Dissemination in Disruption-Tolerant Networks,” Proc. IEEE INFOCOM ’11, 2011.

[21] A. Keranen, J. Ott, and T. Karkkainen, “The ONE Simulator for DTN Protocol Evaluation,” Proc. Second Int’l Conf. Simulation Tools and Techniques (SIMUTools ’