Journal of Computer and System Sciences 80 (2014) 903–915

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Message matching-based greedy behavior detection in delay tolerant networks

Yongming Xie, Guojun Wang *

School of Information Science and Engineering, Central South University, Changsha, Hunan Province, 410083, PR China

ARTICLE INFO

Article history:
Received 25 September 2012
Received in revised form 15 March 2013
Accepted 27 August 2013
Available online 11 February 2014

Keywords:
Delay tolerant networks
Greedy behavior
Mobile trusted module

ABSTRACT

Delay tolerant networks (DTNs) are resource-constrained networks where messages are relayed in a store–carry–forward fashion. In most routing protocols of DTNs, each node is required to honestly relay messages. However, some nodes may violate this principle, and relay messages in a greedy way in order to maximize their own benefit. The existing security solutions in DTNs cannot cope with the greedy behavior because of the greedy nature of nodes. We propose a message matching-based detection (MMBD) method where a smart mobile trusted module (MTM) is introduced to monitor the forwarding sequence of messages buffered in the node to prevent the greedy behavior. This method requires less computation time and fewer resources than the trusted computing group attestation. Furthermore, as shown in simulation results, this method can offer a fair routing environment in DTNs, and it can increase the average message delivery ratio of the network when the greedy behavior occurs.

1. Introduction

As an emerging research area, delay tolerant networks (DTNs) [1] have attracted a lot of attention. In a DTN, a continuous end-to-end connection from a source to a destination is not available because the network is frequently partitioned, and thus messages are relayed in a store–carry–forward fashion. DTNs are usually used to implement communications in some extremely challenging environments where traditional networks may not work, such as interplanetary internet (IPN) [2], vehicular disruption-tolerant networks, US navy seaweb [3], and sparse ad hoc networks.

In most routing protocols of DTNs, each node is required to honestly carry out a specific routing protocol. Generally speaking, a DTN node has constrained resources, such as limited buffer, restricted battery power, and low network bandwidth. Originating from the greedy nature and potential benefit, some nodes are likely to relay messages in a way that maximizes their own benefit. For example, greedy nodes may give a message forwarding priority to each other; or prefer to relay messages with more rewards, such as in credit-based incentive schemes. The greedy behavior impairs the benefit of honest nodes, and undermines the routing fairness of DTNs. Furthermore, this misbehavior decreases the average message delivery ratio of the network.

The greedy behavior in DTNs differs from selfish behaviors where selfish nodes are unwilling to cooperate with members in a DTN. Greedy nodes are ready to participate in the network cooperation, but how to cooperate is decided

* Corresponding author.
E-mail address: csgjwang@csu.edu.cn (G. Wang).

http://dx.doi.org/10.1016/j.jcss.2014.02.001
0022-0000/

by greedy nodes instead of a specific routing protocol. The existing studies on selfish behaviors focus on reputation-based schemes [8–12], and credit-based incentive schemes [13–16]. The reputation-based schemes rely on some special nodes, such as watchdogs and pathraters to detect and isolate misbehaving nodes. The credit-based incentive schemes use credits (virtual coins) to stimulate participants to cooperate. Both reputation-based schemes and credit-based incentive schemes cannot cope with the greedy behavior due to two reasons. First, the greedy behavior often occurs inside a greedy node, which is difficult to be detected by other nodes. Second, greedy nodes aim to maximize their own benefit. Credit-based incentive schemes cannot guarantee the maximization of the benefit of greedy nodes even though they use credits to stimulate cooperation among nodes.

In this paper, we propose a message matching-based detection method which uses a smart mobile trusted module (MTM) [4,5] to detect the greedy behavior. The MTM in a node loads the same routing protocol with the node, and it independently computes a forwarding sequence (S_1) of messages buffered in the node as the routing protocol. Any greedy node who tries to maximize its own benefit surely breaks the routing protocol so that the actual forwarding sequence (S_2) of messages in the node is changed. Thus, the MTM can detect the greedy behavior by matching S_1 with S_2. The MTM will sign messages whose sequence forwarded by the node is consistent with S_1, and messages without the MTM's signature will be discarded by honest receivers. We stress that it may be impracticable for an MTM in DTNs to directly detect misbehaviors inside a node by monitoring the node's system running state and reporting its internal configuration, such as trusted computing group (TCG) attestation and trustworthy computing [6,7], because the node is reluctant to open interfaces and channels for the MTM if its privacy is considered. The contribution of this paper is summarized as follows:

1. We introduce a trusted sublayer containing a smart mobile trusted module to the trusted bundle layer model in order to enhance the security of the DTNs architecture.
2. To simplify the greedy behavior detection and reduce the resources overhead in a DTN node, a message matching-based detection method built on the new trusted bundle layer model is proposed.
3. We present three kinds of greedy behaviors, and discuss the effect of these misbehaviors on the routing fairness and the average message delivery ratio of the network.

Organization. The remainder of this paper is organized as follows: Section 2 overviews related work. In Section 3, we describe a network model and three kinds of greedy behaviors. A new trusted bundle layer model is designed in Section 4. The greedy behavior detection method is presented in Section 5. The simulation and performance evaluation are shown in Section 6. At last, the conclusion is given in Section 7.

2. Related work

Researchers have proposed several schemes to cope with misbehaviors in DTNs. In this section, we overview the existing work and discuss some potential problems.

To handle misbehaving nodes, reputation-based schemes were first considered by previous studies [8,9]. Marti et al. proposed a dynamical measurement method [8] which introduces two kinds of special nodes, watchdogs and pathraters. Watchdogs are used to identify misbehaving nodes, and the function of pathraters is to help routing protocols to avoid misbehaving nodes. Later, Buchegger and Boudec [9] presented a similar approach called CONFIDANT. We argue that these reputation-based schemes may not be suitable for the greedy behavior because it is impractical to detect or identify greedy nodes in an extremely challenging environment. Furthermore, how to propagate the reputation of misbehaving nodes throughout a large delay network is still an open issue.

The credit-based incentive schemes [13–16] provide another solution, which adopt credits (virtual coins) to stimulate selfish nodes to cooperate rather than detect misbehaving nodes. The credit-based incentive schemes are similar to an electronic cash system, where a virtual bank (VB) takes charge of credits management and cash clearance. To compensate resources consumption, a source pays a certain number of credits to intermediates. If the credits of a source are not enough to send a message, the source has to get more credits by relaying the messages of other nodes. This kind of incentive schemes can efficiently deal with selfish behaviors, but they are not applicable to deal with the greedy behavior as argued in Section 1.

The third category of solutions is tit-for-tat based incentive schemes [17,18], where a node autonomously lowers the quality of service to misbehaving neighbors, and fully cooperates with honest neighbors. These mechanisms may temporarily isolate misbehaving nodes, but they still suffer a bootstrapping problem. In [19], Shevade et al. developed an incentive-aware routing protocol that incorporates 'generosity' and 'contribution' to address the above issue. However, this approach needs to generate a set of candidate paths from a source to a destination, and then approximates the message delivery ratio using the linear programing optimization. Unfortunately, it is difficult to determine a path before routing since DTNs follow the opportunistic routing.

Our method employs an MTM to detect and inhibit the greedy behavior inside a node by matching the forwarding sequences of messages between the MTM and the node, and it is different from three classes of schemes above, which depend on other nodes, such as watchdogs, pathraters, VB, and neighbors.

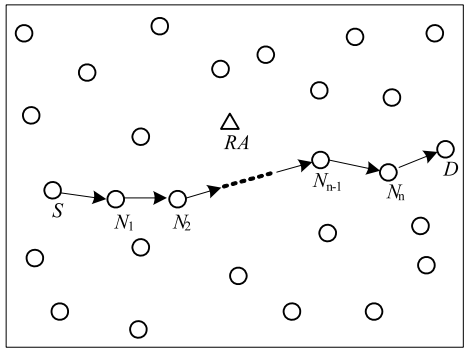

Fig. 1. Network model.

3. Network model and greedy behaviors

We first describe a network model, and then discuss three kinds of greedy behaviors which may happen in DTNs.

3.1. Network model

We formalize a DTN as a set of mobile devices held by individuals. Here, we use a node N_i to denote a mobile device. Each node has constrained wireless network resources, such as limited buffer, restricted battery power, and low network bandwidth. Because of these constrained resources, the node is difficult to construct a continuous end-to-end connection. As shown in Fig. 1, a source S transmits messages to a destination D through n intermediates after a large propagation delay. Additionally, to prevent unauthorized nodes from accessing the DTN [20], we assume that there exists a management authority referred to as registration authority (RA) in this paper.

3.2. Greedy behaviors

All the nodes in DTNs implement the same routing protocol which is specified initially, but some nodes breach the routing protocol, and relay messages in their own way in order to exploit fewer resources to maximize their own benefit. We present some greedy behaviors.

1. **Greedy Behavior I**. Greedy nodes give a message forwarding priority to each other, and they attempt to get more transmission opportunities. But, honest nodes suffer an unfair treatment so that their message delivery ratio is lower.
2. **Greedy Behavior II**. Most reward forms in credit-based incentive schemes depend on the length (hops) of a message delivery path. The shorter the path is, the more credits the intermediates on a message delivery path get. Greedy intermediates set a hops threshold ϵ, and prefer to relay a message whose current hops are less than ϵ. It is unfair for messages and honest intermediates on a longer path.
3. **Greedy Behavior III**. DTNs are resource-constrained networks. Roughly speaking, messages which require larger relay time will consume more resources of intermediates. To save their own resources, greedy nodes preferentially relay messages that require smaller relay time, but these messages may not be consistent with messages decided by the routing protocol. For example, in the encounter probabilistic routing protocol (Prophet) [21], a message is forwarded to an intermediate who has higher encounter probability with the destination of the message. A greedy intermediate predicts time consumption of relaying a message according to the last encounter time and the encounter frequency with the destination. If the relay time is larger, the intermediate does't relay the message even though it has higher encounter probability with the destination.

Greedy nodes like to relay messages which can maximize their own benefit, and these messages are called as interested messages. On the other hand, most messages fail to attract the interest of greedy nodes, and suffer an unfair treatment so that a great number of transmission opportunities are seized by interested messages. We refer to these messages as innocent messages. Apart from the routing unfairness, another serious issue is that these greedy behaviors decrease the average message delivery ratio of the network. Generally, a well-designed routing protocol has higher message delivery ratio. Greedy behaviors may increase their own message delivery ratios, but they will affect the overall performance of the network.

- In Greedy Behavior I, greedy nodes don't fully cooperate with honest nodes so that total transmission opportunities in the network are decreased. Correspondingly, the average message delivery ratio of the network is degraded.

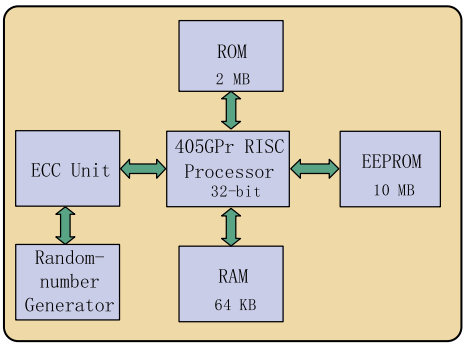

Fig. 2. Mobile trusted module based on PCIeCC.

- To get more credits, greedy intermediates in Greedy Behavior II prefer to relay a message which has been relayed by fewer nodes, but most messages are in the longer path, and they hardly get transmission opportunities from greedy intermediates. The average message delivery ratio of the network is lower because of this misbehavior.
- Prophet is an efficient forwarding-based routing protocol, and has higher message delivery ratio than other forwarding-based routing protocols. However, nodes in Greedy Behavior III use the relay time as metric instead of the encounter probability to relay messages. This misbehavior impairs the performance of this protocol.

4. Trusted bundle layer architecture

4.1. Mobile trusted module

With increasing threats to network security, a trusted platform module (TPM) is extensively applied in Internet to satisfy security and privacy requirements. The TPM is not applicable for mobile networks because the mobility leads to unplanned interactions between mobile devices. Thereby, the trusted computing group (TCG) developed an open security specification called mobile trusted module (MTM), such as tamper-proof secure module which has been adopted to solve some security problems in DTNs and ad hoc networks [13,22]. However, emerging reports on TPM attacks [23,24] may discourage the confidence in the MTM, and it seems to be difficult to implement a fully secure MTM. But, the MTM is reliable in our method considering the following factors:

1. As argued in [13], researchers and manufactures are steadily improving the security of an MTM. For example, the latest IBM PCIe Cryptographic Coprocessor (PCIeCC) [25] provides a high-security and high-throughput cryptographic subsystem. It is very difficult for attackers to conquer an MTM in limited time or with restricted resources.
2. Our method is merely used to cope with greedy behaviors rather than to encrypt a user's confidential messages. As for attackers, it is not worth expending a large mount of money and time to design a special attack module.

To satisfy the requirement of mobile devices, we use a smart MTM structure referred to PCIeCC as shown in Fig. 2. The MTM includes a 32-bit IBM RISC processor, 64 KB RAM, 2 MB ROM, 10 MB EEPROM, an elliptic curve cryptography (ECC) unit, and a random-number generator. The ECC unit is a critical component of the MTM, which serves for key generation, digital signature, and verification. The MTM is desired to be portable and can be fixed easily like a SIM card in a mobile phone. Any node (N_i) that joins a DTN must be equipped with an MTM (M_i). Here, N_i is called as M_i' host. The functions of an MTM are described as follows:

- Collect routing information from interactions among neighbors.
- Record relay requests of messages.
- Compute a forwarding sequence of messages when a connection opportunity is available.
- Check whether messages from the MTM's host are consistent with the messages in the forwarding sequence, and sign these messages if the check holds.

4.2. MTM-based bundle layer model

The existing DTNs routing protocols are based on the bundle layer model [26] including application layer, bundle layer, transport layer, network layer, and MAC/PHY layer. We modify the bundle layer model, and add a trusted sublayer into MAC layer as illustrated in Fig. 3. The application layer, bundle layer, transport layer, network layer, and MAC layer are called as the upper layers of the trusted sublayer. As usual, an MTM is embedded into this sublayer, and detects the greedy behaviors of nodes. The new bundle layer model is called as a trusted bundle layer model which opens two data channels:

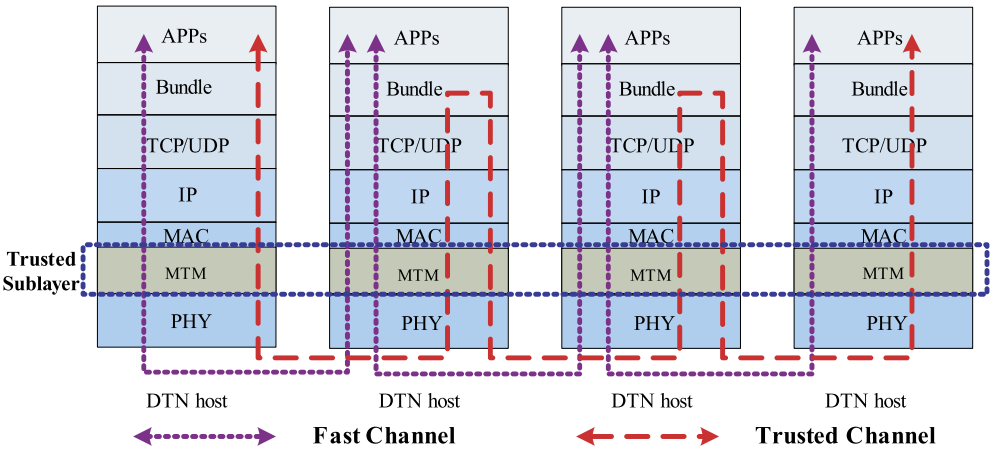

Fig. 3. Trusted bundle layer model.

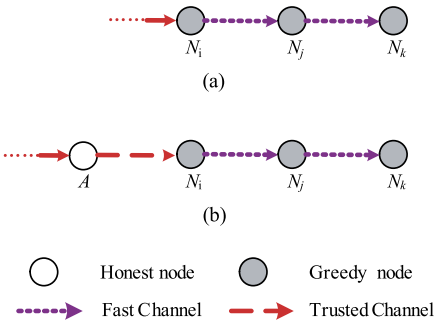

Fig. 4. (a) The misbehavior exploiting the fast channel; (b) Detection method to the misbehavior in (a).

fast channel and trusted channel, and the corresponding transmitted messages are direct-delivery messages and in-transit messages. Note that messages appeared in other sections denote the in-transit messages by default. The fast channel directly delivers messages destined for neighbors. The MTM will not inspect direct-delivery messages sent by the host so as to save resources. The trusted channel is used to relay in-transit messages that are detected by the MTM.

Although the fast channel facilitates the interactions between neighbors, it may bring about an additional trouble. To maximize their own benefit, some greedy nodes may take advantage of the fast channel to relay in-transit messages, but the MTMs loaded in these greedy nodes would fail to perceive this misbehavior. For example, greedy nodes N_i, N_j and N_k collaborate with each other as described in Fig. 4(a). When N_i receives an in-transit message (p) destined for N_k, it will transform p into a direct-delivery message (p') in order to achieve Greedy Behavior I. N_i replaces p's destination IP address with N_j' address, and it may even change the endpoint IDs in the bundle block format. Meanwhile, the original destination IP address and endpoint IDs are stored in the bundle payload. N_i sends p' to N_j through the fast channel. N_j restores p' to the original in-transit message p, and delivers p to N_k via the trusted channel. We discuss the probability P of this misbehavior. Let m be the total number of nodes and n be the number of greedy nodes. The notation l denotes an average path length, P_c means the probability of an opportunistic connection between two greedy nodes, and P_t is the transmission probability. Thus, $P = (\frac{n}{m} P_c P_t)^l$. The probability of this misbehavior is roughly negligible because $\frac{n}{m}$, P_c, and P_t are usually very small. Additionally, the unpredictable nature of DTNs reduces the effectiveness of this misbehavior [27].

The trusted sublayer improves the security of the bundle layer model, and can efficiently detect greedy behavior inside a node.

5. Message matching-based detection

In this section, we first propose a basic message matching-based detection (MMBD) method which aims to detect and inhibit the greedy behavior, and then we propose an optimized this MMBD method to reduce the computation and resources overhead in a DTN node.

5.1. Basic MMBD

The basic MMBD method consists of system initialization, information collection, and greedy behavior detection.

5.1.1. System initialization

Our work is based on a bilinear pairing over elliptic curves. Let $\mathbb{G}$ be an additive cyclic group, and $\mathbb{G}_T$ be a multiplicative cyclic group with the same prime order q. P is a generator of $\mathbb{G}$. We assume that e is a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, which satisfies the following properties:

1. Bilinearity: For $a, b \in \mathbb{Z}_q$, $e(aP, bP) = e(P, P)^{ab}$.
2. Non-degeneracy: $e(P, P) \neq 1_{\mathbb{G}_T}$.
3. Computability: There is an efficient algorithm to compute $e(R, S)$ for any $R \in \mathbb{G}$ and $S \in \mathbb{G}$.

At the initial stage, the registration authority in the network generates system parameters $\{q, \mathbb{G}, \mathbb{G}_T, e, P\}$ and a hash function: $H : \{0, 1\}^* \to \mathbb{G}$, which are preloaded in each DTN node. A node N_i randomly chooses s_i as its private key that corresponds to a public key, $PK_i = s_i P$. Let $Enc(*)$ be an encryption algorithm, and $Dec(*)$ be the corresponding decryption algorithm. Accordingly, an MTM (M_i) in N_i possesses an independent private key λ_i and a public key, $PM_i = \lambda_i P$. $Sign(*)$ and $Sigm(*)$ are signature functions of N_i and M_i, respectively.

5.1.2. Information collection

When a connection opportunity approaches, nodes exchange some relay information between neighbors, which includes a connection event, routing information, and a relay request of messages. A connection event means that neighbors have successfully constructed a channel connection and are preparing to deliver messages. The connection event will be handed over all the layers in a node. The routing information contains neighbors' knowledge on the network environments, such as encounter history, node mobility track, and reputation. The node will use the routing information to make a routing decision. A relay request describes the basic information of messages, such as IP addresses, endpoint IDs, TTL, sizes of messages, and time stamp. However, a greedy node can still misbehave as described in Section 3 if the MTM in the node fail to get the relay information. Thus, the MTM must obtain the relay information prior to the upper layers as follows:

1. The PHY layer perceives the approaching neighbors, and then a connection event is generated and handed over all the layers.
2. When nodes N_i and N_j have constructed a connection, N_i generates routing information and a relay request (R) which are encrypted by M_i, $C = Enc_{PM_j}(R)$.
3. M_j in N_j decrypts the ciphertext C, $R = Dec_{\lambda_j}(C)$. M_j saves the routing information, and records the relay request if these messages are successfully received by the node.

5.1.3. Greedy behavior detection

The existing detection approaches to monitor misbehaviors inside a node are TCG attestation and trustworthy computing [28], including hardware-based attestation and software-based attestation. The attestation is used to monitor system running state and report internal configuration, such as software components (BIOS, bootloader, applications) measurement, buffer content verification, and operating system attestation. But, the attestation may not be suitable for DTNs. First, these approaches are mainly developed for non-resource constrained computer systems, and require each communication peer to exchange verifying entities. Second, mobile devices may be unwilling to open interfaces and channels for direct control access when their privacy and security are concerned. So, we adopt another simple method, message matching-based detection, which detects misbehaviors by matching the forwarding sequences of messages rather than measuring the system configuration.

In this paper, each MTM loads the same routing protocol with its host in advance. The current routing protocols in DTNs are not complex, and they can be implemented by an MTM. The MTM is driven by connection events. When a connection event (CE) arrives at the trusted sublayer, the MTM runs the routing protocol according to collected routing information and relay requests above, and then generates next forwarded message via the available connection. Meanwhile, the basic information (P_n) of this message is abstracted by the MTM, P_n: *source IP*$\|$*destination IP*$\|$*endpoint IDs*$\|$*sizes*$\|$*time stamp*.

This connection event is handed over until the bundle layer. The host also decides next forwarded message $(\mathcal{P})$, and attaches $\mathcal{P}$'s signature $(Sign(\mathcal{P}))$ on $\mathcal{P}$. When $\mathcal{P}$ is handed down to the trusted sublayer through the trusted channel, the MTM extracts $\mathcal{P}$'s basic information (P_b), and matches P_n with P_b by checking whether the following equation holds or not:

$$H(P_b) \stackrel{?}{=} H(P_n) \tag{1}$$

It indicates that the host is honest if Eq. (1) holds, so the MTM will sign $\mathcal{P}$ with its private key $(Sigm(\mathcal{P}))$ and send it as the message format depicted in Fig. 5(a). Otherwise, the MTM believes that the host is a greedy node, and refuses to sign $\mathcal{P}$. Obviously, the forwarding sequence of messages in Greedy Behavior I, II and III is inconsistent with the forwarding sequence in the MTM.

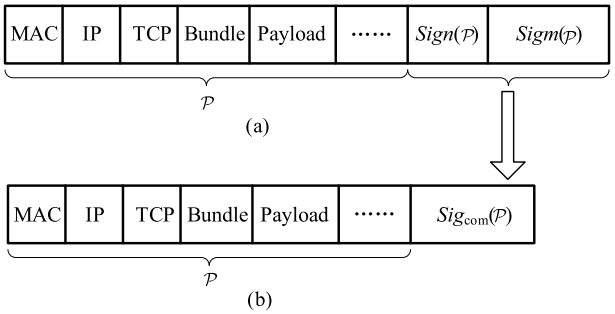

Fig. 5. (a) The message format in the basic MMBD method; (b) The message format in the optimized MMBD method.

In particular, a greedy node may misbehave as Section 4.2 and delivers $\mathcal{P}$ by the way of a fast channel. As for this misbehavior, we merely concern the first greedy node N_i whose previous hop is an honest node (A) illustrated in Fig. 4(b). M_i records the relay request from A. If N_i misbehaves, the normal forwarding sequence of messages in N_i will be changed. The sequence cannot match with the forwarding sequence in the MTM, so the MTM can perceive N_i' misbehavior. The specific detection algorithm is depicted in Algorithm 1.

Algorithm 1 Basic MMBD algorithm.

```
Store available connection opportunities in a queue CO;
Collect the routing information saved in RI;
Record the relay request in a queue RR;
while Size(CO) > 0 do
    l ← GetConnection(CO);
    p ← ComputeCandidate(l, RR, RI);
    Add(Q, p);
end while
P_n ← GetCandidate(Q);
P ← Wait();
P_b ← Extract(P);
if Match(P_b, P_n) then
    Sigm(P);
else
    Refuse to sign P;
end if
```

Variable:

Q: candidate queue.

Macros:

GetConnection(CO): get a specific connection from CO;
ComputeCandidate(l, RR, RI): generate a candidate message from RR;
Add(Q, p): add p to the end of Q;
Sigm(p): MTM signs for $\mathcal{P}$;
GetCandidate(Q): get a candidate from Q;
Wait(): wait for a message $\mathcal{P}$ from the upper layers;
Extract($\mathcal{P}$): extract the information of the message $\mathcal{P}$;
Match(P_b, P_n): check whether P_b is consistent with P_n.

5.2. Optimized MMBD

In the aforementioned detection method, a message needs to be matched by the MTM, and is signed twice by the host and the MTM, which may lead to a larger delay and unnecessary resources overhead. Here, we will improve the efficiency of the basic MMBD method, and combine *Sign*($\mathcal{P}$) with *Sigm*($\mathcal{P}$). Meanwhile, we guarantee that the combination signature should achieve the same function with the basic MMBD algorithm.

In the optimized MMBD method, an MTM assumes that its host is honest during a period T_j, and grants a trust certificate (a signature on T_j) to the host. If the host relays messages as the routing protocol, the MTM will update the trust certificate at the next period T_{j+1}, and vice versa. The messages relayed by the host without a trust certificate will not be received by honest nodes. Thus, the trust certificate can restrain the greedy behaviors of the node, which is similar to a single signature in the basic MMBD method. The optimized detection method based on the trust certificate is presented as follows:

1. M_i periodically gives a trust certificate about T_j, $Cert_j = \lambda_i H(T_j)$ to N_i.
2. N_i decides a forwarded message ($\mathcal{P}$), and signs it, $Sign(\mathcal{P}) = s_i H(\mathcal{P})$. Then, N_i constructs a combination signature $Sig_{com}(\mathcal{P}) = Sign(\mathcal{P}) + Cert_j + s_i H(T_j)$.
3. N_i attaches $Sig_{com}(\mathcal{P})$ on $\mathcal{P}$ as shown in Fig. 5(b), and hands down $\mathcal{P}$ to the trusted sublayer.
4. M_i saves the basic information (P_b) of $\mathcal{P}$ and directly submits it to the PHY layer, and P_b will be checked by M_i later. If M_i finds that $H(P_b) \neq H(P_n)$, it will refuse to give a new trust certificate to N_i at the next period T_{j+1}.
5. When $\mathcal{P}$ arrives at the receiver (N_j), M_j does nothing but piggybacks M_i' public key (PM_i).
6. N_j's bundle layer verifies $Sig_{com}(\mathcal{P})$ by checking whether the following equation holds or not:

$$
\begin{aligned}
e\big(Sig_{com}(\mathcal{P}), P\big) &\stackrel{?}{=} e\big(H(T_i), PK_i + PM_i\big)e\big(H(\mathcal{P}), PK_i\big) \qquad (2)\\
e\big(Sig_{com}(\mathcal{P}), P\big) &= e\big(s_i H(T_i) + \lambda_i H(T_i) + s_i H(\mathcal{P}), P\big)\\
&= e\big(s_i H(T_i) + \lambda_i H(T_i), P\big)e\big(s_i H(\mathcal{P}), P\big)\\
&= e\big((s_i + \lambda_i)H(T_i), P\big)e\big(s_i H(\mathcal{P}), P\big)\\
&= e\big(H(T_i), (s_i + \lambda_i)P\big)e\big(H(\mathcal{P}), s_i P\big)\\
&= e\big(H(T_i), PK_i + PM_i\big)e\big(H(\mathcal{P}), PK_i\big)
\end{aligned}
$$

If the check passes, the receiver believes that the sender is a trustworthy node during T_i, and then relays the message. The optimized detection method combines a host's signature with an MTM's signature, and periodically updates a trust certificate, so it requires less signature and verification time, and reduces the message delivery delay. The specific optimized MMBD algorithm as shown in Algorithm 2.

Algorithm 2 Optimized MMBD algorithm.

```
1: Let the current period be T_j.
2: Store available connection opportunities in a queue CO;
3: Collect the routing information saved in RI;
4: Record the relay request in a queue RR;
5: while Size(CO) > 0 do
6:    l ← GetConnection(CO);
7:    p ← ComputeCandidate(l, RR, RI);
8:    Add(Q, p);
9:    if isPeriod(T_j) then
10:      if HS_{j-1} == True&&F_j == False then
11:         Cert_j ← Sigm(T_j);
12:         SendToUpperLayer(Cert_j);
13:         F_j ← True;
14:         F_{j+1} ← False;
15:      end if
16:   end if
17: end while
18: P_n ← GetCandidate(Q);
19: P ← Wait();
20: P_b ← Extract(P);
21: SendToPHY(P);
22: if notMatch(P_b, P_n) then
23:    HS_j ← False.
24: end if
```

Variables:

HS_{j-1}: it is true if the host is honest during T_{j-1};
F_j: update flag of a trust certificate;

Macros:

SendToUpperLayer($Cert_j$): submit a trust certificate on T_j to the upper layers;
GetConnection(CO): get a specific connection from CO;
ComputeCandidate(l, RR, RI): generate a candidate message from RR;
Add(Q, p): add p to the end of Q;
Sigm(p): MTM signs for $\mathcal{P}$;
GetCandidate(Q): get a candidate from Q;
Wait(): wait for a message $\mathcal{P}$ from the upper layers;
notMatch(P_b, P_n): check whether P_b is not consistent with P_n;
Extract($\mathcal{P}$): extract the information of the message $\mathcal{P}$;
SendToPHY($\mathcal{P}$): send $\mathcal{P}$ to PHY layer.

Table 1
Cryptographic operations test.

Operation	Description	Execution time
T_{pmul}	Point multiplication in $\mathbb{G}$	0.55 ms
T_{pair}	Bilinear pairing operation	2.28 ms
T_{padd}	Point addition	0.004 ms

Table 2
Computation overhead.

Message signature	Combination signature	Message verification
T_{pmul}	$2 \times T_{padd} + T_{pmul}$	$3 \times T_{pair} + T_{padd}$

Table 3
Simulation parameters.

Parameter	Value
Simulation time	12 hours
Number of nodes	100 nodes
Transmission range	50 m
Transmission speed	250 KB/s
Mobile speed	0.5–1.5 m/s
Mobility model	Map-based mobility model
Application protocol	PingApplication
Routing protocol	Epidemic, Prophet
Message size	500 KB
Message generation interval	100 s
Combination signature period	600 s

6. Overhead and performance evaluation

6.1. Computation overhead

The optimized MMBD method needs to perform many cryptographic operations built on the elliptic curve cryptography (ECC), including bilinear pairing, point multiplication, point addition, and hash function. We use a machine with Intel 3.0 GHz CPU and 1 GB RAM to measure the computation time of these operations, based on the cryptographic library MIRACL.[1] The results are shown in Table 1.

In the optimized MMBD method, an MTM periodically update the trust certificate, and thus the computation overhead incurred by the MTM is negligible relative to its host. In reality, the most computation overhead of a host arises from the message signature and verification, which is summarized in Table 2. The total computation overhead of a message is $3 \times T_{pair} + 3 \times T_{padd} + 2 \times T_{pmul}$.

6.2. Simulation settings

We adopt the opportunistic network environment (ONE) simulator[2] which is a powerful simulation tool. ONE supplies a lot of dominant DTN routing protocol implementations, such as MaxProp, SprayAndWait, Prophet, and Epidemic. The simulations choose Epidemic and Prophet as routing protocols, and use the default setting of ONE_1.4.0 where each node has the same resources configuration as described in Table 3. We implement three paradigms for Greedy Behavior I, II and III as follows:

1. **Paradigm I**. 10 nodes are arranged to act as a greedy group in Greedy Behavior I, where members give a message forwarding priority to each other. However, the messages from other 90 honest nodes cannot be treated fairly.
2. **Paradigm II**. In most credit-based incentive schemes, the rewards depend on the length of a message delivery path. We assume 10 nodes misbehave like Greedy Behavior II, and the hops threshold ϵ is set to 3 in the paradigm. When a message has been relayed by $\epsilon - 1$ intermediates, greedy nodes will refuse to receive the message.
3. **Paradigm III**. We deploy 10 nodes to implement Greedy Behavior III. These greedy nodes use the last encounter time and the encounter frequency to predict the relay time of a message. In our simulations, the average relay time (t) of a message in a fair environment is about 1100 s. To save its own resources, the greedy nodes merely relay messages

[1] Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL), http://certivox.com/.
[2] http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

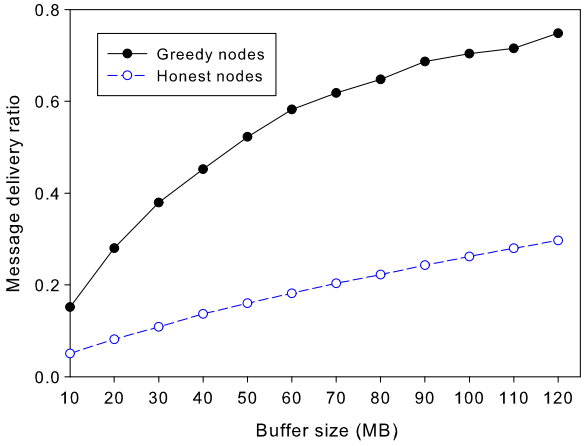

Fig. 6. The average message delivery ratio comparison of greedy nodes with honest nodes in Paradigm I.

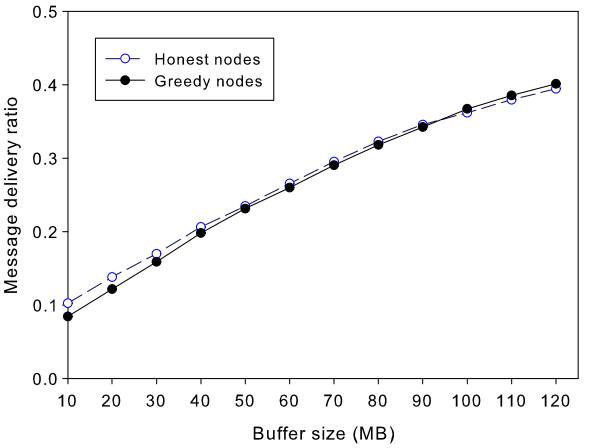

Fig. 7. The average message delivery ratio comparison of greedy nodes with honest nodes in MMBD.

whose predicted relay times are less than $0.5t$, e.g., 550 s. Although these messages consume less relay time, these greedy nodes don't relay extra messages during the simulation if their credits are enough.

We compile an MTM simulation program to realize functions of an MTM described in Section 4. When nodes misbehave during period T_j, the MTM simulator will refuse to update a trust certificate for these nodes at the next period T_{j+1}. Here, the period is 600 s, and the frequency of misbehavior implementation is 1200 s.

6.3. Routing fairness

1. In Paradigm I, greedy nodes can get more transmission opportunities, and messages from greedy nodes have higher average delivery ratio than honest nodes. But, messages from honest nodes only get fewer opportunities so that these messages have lower average delivery ratio. Fig. 6 illustrates the result, the average message delivery ratio of greedy nodes reaches 0.541, and the corresponding ratio of honest nodes is only 0.187. Furthermore, the delivery ratio increment of greedy nodes from 10 MB to 120 MB is also higher than honest nodes. Contrarily, MMBD can efficiently inhibit this misbehavior, and provides a fair network environment. As shown in Fig. 7, greedy nodes and honest nodes have similar average message delivery ratio, and their ratios are 0.248 and 0.243, respectively.
2. Most credit-based incentive schemes, such as SMART [16], require that nodes fairly relay all the messages. To gain more credits with fewer resources in Greedy Behavior II, greedy nodes break the SMART scheme, and only relay a message whose hops are less than 3. Therefore, these nodes can gain more credits than honest nodes since the rewarded credits are decided by length (hops) of a message delivery path. We use a reward ratio to evaluate this kind of misbehavior. The reward ratio is the proportion of total rewarded credits to the number of relayed messages. Fig. 8 displays that greedy nodes have higher reward ratio than honest nodes. The ratio means that greedy nodes can gain excess rewards even though they relay the same number of messages with honest nodes. The average reward ratio of greedy nodes is 0.523, but it is merely 0.258 for honest nodes. The proposed method restrains this misbehavior. As illustrated in Fig. 9, the average reward ratio of greedy nodes is degraded into 0.313 which is similar to 0.316 of honest nodes in MMBD.
3. Nodes in Greedy Behaviors III are only interested in messages whose predicted relay time is less than 550 s, the average relay time of message in these nodes is approximately 423 s as shown in Fig. 10. Contrarily, honest nodes relay messages according to the encounter probability in Prophet, and don't concern the relay time. So, the average relay time in honest nodes is larger than greedy nodes, and it reaches over 1180 s. This result indicates that honest nodes consume more resources than greedy nodes per relaying a message. In our method, it is difficult for greedy nodes to implement this misbehavior. The average relay time in greedy nodes nearly equals the time in greedy nodes as shown in Fig. 11, and they are 1102 s and 1091 s, respectively.

6.4. Average message delivery ratio

1. The proposed method mitigates the undesirable effect of greedy nodes on the average message delivery ratio of the network in Paradigm I. As shown in Fig. 12, the maximum message delivery ratio in Paradigm I is only 0.342, which is lower than 0.364 in Epidemic. MMBD is able to achieve the same message delivery ratio with Epidemic, and their curves are almost identical.

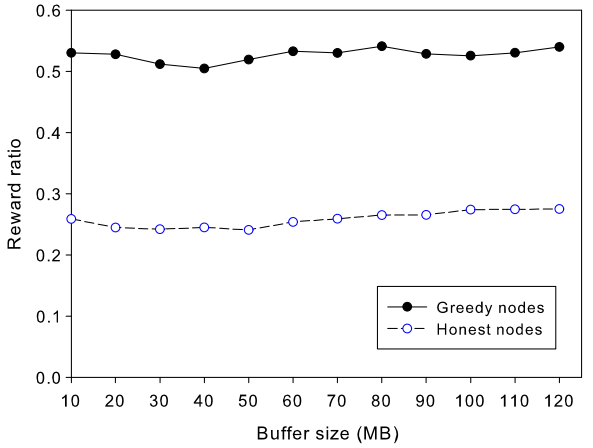

Fig. 8. The average reward ratio comparison of greedy nodes with honest nodes in Paradigm II.

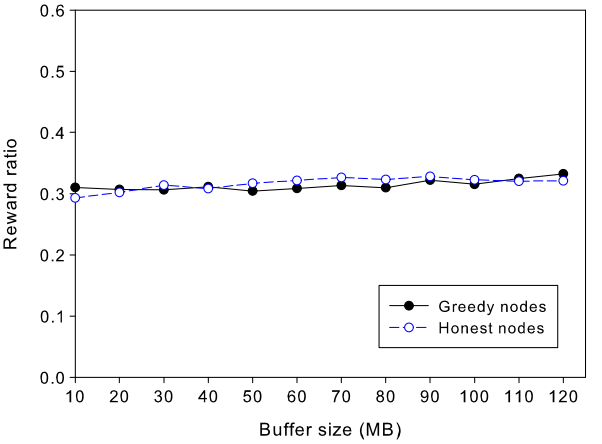

Fig. 9. The average reward ratio comparison of greedy nodes with honest nodes in MMBD.

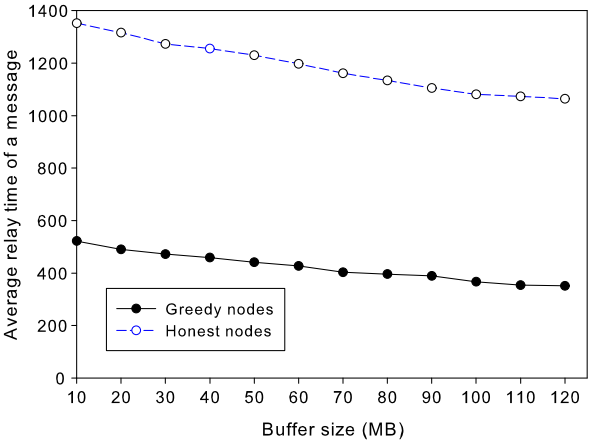

Fig. 10. The average relay time comparison of greedy nodes with honest nodes in Paradigm III.

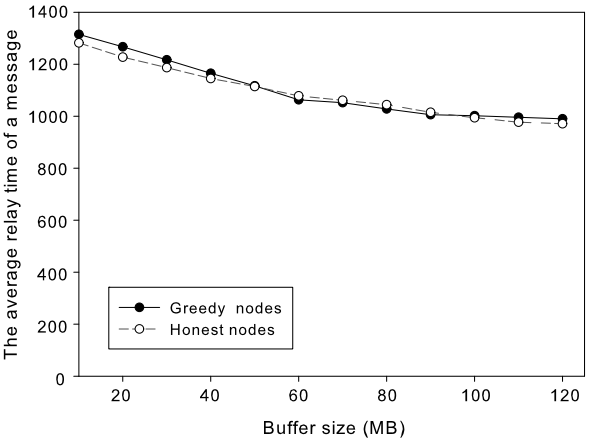

Fig. 11. The average relay time comparison of greedy nodes with honest nodes in MMBD.

2. Greedy nodes refuse to relay messages whose hops are more than 2, and the average message delivery ratio of the network is 0.209 as shown in Fig. 13. Our method enhances the ratio in comparison with Paradigm II, and it is 0.235 which is equivalent to Epidemic.
3. The average message ratio of the network in Paradigm III is lower because greedy nodes violate the original routing protocol. The maximum message delivery ratio is only 0.251 when the buffer size is 120 MB. In our method, the MTM simulator prevents this misbehavior so that the ratio is increased to 0.375 under the same buffer sizes in Fig. 14.

7. Conclusion

Most routing protocols in DTNs require each node to honestly relay messages, but a few greedy nodes violate this principle in order to maximize their own benefit. This greedy behavior breaks the routing fairness and decreases the message delivery ratio of DTNs. In this paper, we introduced a trusted sublayer containing a smart mobile trusted module (MTM) to the bundle layer model, and proposed a message matching-based detection method, which uses an MTM to detect this greedy behavior by monitoring the forwarding sequence of messages inside a node. Therefore, it requires less computation time and fewer resources than the trusted computing group attestation. Furthermore, it can offer a fair routing environment for honest nodes in DTNs, and increases the average message delivery ratio of the network when the greedy behavior occurs.

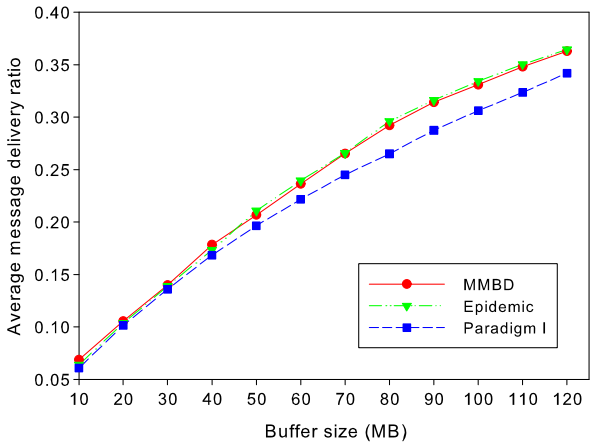

Fig. 12. The average message delivery ratios in Paradigm I, Epidemic, and MMBD.

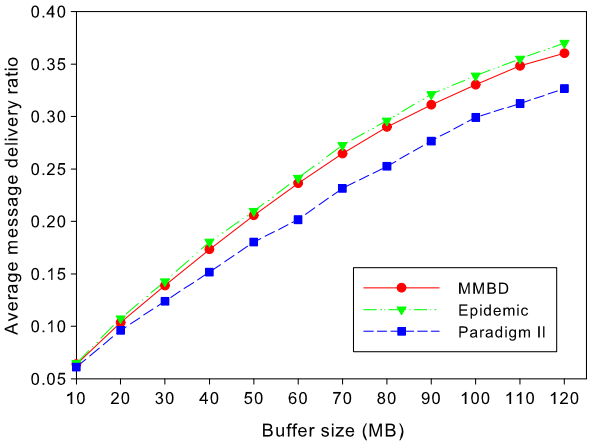

Fig. 13. The average message delivery ratios in Paradigm II, Epidemic, and MMBD.

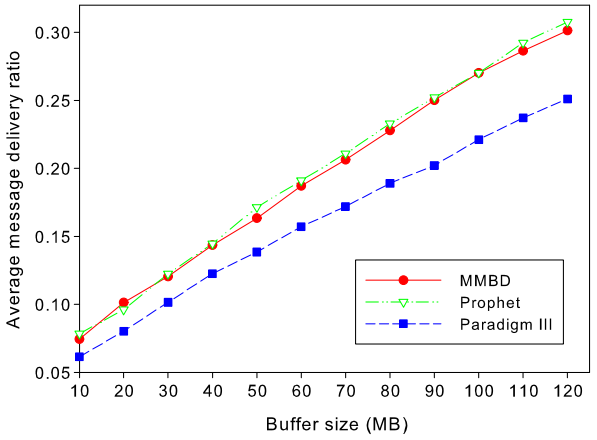

Fig. 14. The average message delivery ratios in Paradigm III, Prophet, and the proposed method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant numbers 61272151, 61073037 and 61103035, and the Ministry of Education Fund for Doctoral Disciplines in Higher Education under grant number 20110162110043.

References

[1] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proceedings of ACM SIGCOMM, August 2003, pp. 27–34.
[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, H. Weiss, Interplanetary Internet (IPN): Architectural Definition, http://www.ipnsig.org/reports/memo-ipnrg-arch-00.pdf.
[3] J.A. Rice, R.K. Creber, C.L. Fletcher, P.A. Baxley, K.E. Rogers, D.C. Davison, Evolution of Seaweb underwater acoustic networking, in: Proceedings of IEEE Conference of the Oceans on Information Systems and Sciences, September 2000, pp. 2007–2017.
[4] TCG Mobile Reference Architecture, Specification v.1.0, revision 1, June 2007.
[5] TCG Mobile Trusted Module, Specification v.1.0, revision 6, June 2008.
[6] Specifications are available on the Trusted Computing Group web site http://www.trustedcomputinggroup.org.
[7] S. Pearson, B. Balacheff, L. Chen, D. Plaquin, G. Proudler, Trusted Computing Platforms: TCPA Technology in Context, 1st edition, ISBN 0130092207, 2002.
[8] S. Marti, T. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proceedings of ACM MobiCom, 2000, pp. 255–265.
[9] S. Buchegger, J. Boudec, Performance analysis of the CONFIDANT protocol: Cooperation of nodes-fairness in dynamic ad-hoc networks, in: Proceedings of IEEE/ACM Workshop on MobiHoc, 2002, pp. 226–236.
[10] G. Dini, A.L. Duca, A reputation-based approach to tolerate misbehaving carriers in delay tolerant networks, in: Proceedings of IEEE Symposium on Computers and Communications, 2010, pp. 772–777.
[11] V. Natarajan, Y. Yang, S. Zhu, Resource-misuse attack detection in delay-tolerant networks, in: Proceedings of IEEE 30th International Performance Computing and Communications Conference, November 2011, pp. 1–8.

[12] W. Peng, F. Li, X. Zou, J. Wu, Behavioral detection and containment of proximity malware in delay tolerant networks, in: Proceedings of IEEE 8th International Conference on Mobile Ad hoc and Sensor Systems (MASS), October 2011, pp. 411–420.
[13] L. Buttyan, J.P. Hubaux, Enforcing service availability in mobile ad-hoc WANs, in: Proceedings of IEEE/ACM Workshop on MobiHoc, August 2000, pp. 87–96.
[14] S. Zhong, J. Chen, Y.R. Yang, Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks, in: Proceedings of IEEE INFOCOM, March 2003, pp. 1987–1997.
[15] S. Lee, G. Pan, J. Park, M. Gerla, S. Lu, Secure incentives for commercial ad dissemination in vehicular networks, in: Proceedings of ACM MobiHoc, 2007, pp. 150–159.
[16] H. Zhu, X. Lin, R. Lu, X. Shen, D. Xing, Z. Cao, An opportunistic batch bundle authentication scheme for energy constrained DTNs, in: Proceedings of IEEE INFOCOM, March 2010, pp. 605–613.
[17] V. Srinivasan, P. Nuggehalli, C. Chiasserini, R. Rao, Cooperation in wireless ad hoc networks, in: Proceedings of IEEE INFOCOM, 2003, pp. 808–817.
[18] J.J. Jaramillo, R. Srikant, DARWIN: Distributed and adaptive reputation mechanism for wireless ad-hoc networks, in: Proceedings of ACM MobiCom, 2007, pp. 87–98.
[19] U. Shevade, H. Song, L. Qiu, Y. Zhang, Incentive-aware routing in DTNs, in: Proceedings of IEEE ICNP, October 2008, pp. 238–247.
[20] H. Samuel, W. Zhuang, Preventing unauthorized messages in DTN based mobile ad hoc networks, in: Proceedings of IEEE GlobeCom, November 2009, pp. 1–6.
[21] A. Lindgren, A. Doria, O. Scheln, Probabilistic routing in intermittently connected networks, in: Proceedings of ACM MobiHoc, 2003.
[22] G. Guette, C. Bryce, Using TPMs to secure vehicular ad-hoc networks (VANETs), in: Proceedings of WISTP, in: Lect. Notes Comput. Sci., vol. 5019, 2008, pp. 106–116.
[23] TPM Reset Attack, http://www.cs.dartmouth.edu/pkilab/sparks, 2007.
[24] Successful Attack on TPM, https://mocana.com/blog/2010/02/09/successful-attack-on-tpm, February, 2010.
[25] IBM PCIe Cryptographic Coprocessor, http://www-03.ibm.com/security/cryptocards/pciecc/overview.shtml.
[26] K. Scott, S. Burleigh, Bundle Protocol Specification, IETF RFC 5050, experimental, November 2007.
[27] J. Burgess, G.D. Bissias, M. Corner, B.N. Levine, Surviving attacks on disruption-tolerant networks without authentication, in: Proceedings of ACM MobiHoc, September 2007, pp. 61–70.
[28] A. Seshadri, M. Luk, E. Shi, A. Perrig, L.V. Doorn, P. Khosla, Pioneer: Verifying integrity and guaranteeing execution of code on legacy platforms, in: Proceedings of ACM Symposium on Operating Systems Principles (SOSP), October 2005, pp. 1–16.