Audit Based Misbehaviour Detection Of Nodes In Wireless Adhoc Network

Mrs. Vrushali M. Bodhankar
M.E [C.S.E] ,
Computer Science and Engg,
DIEMS, Aurangabad

Mrs. Ashwini S. Gaikwad
Assistant Professor
Computer Science and Engg,
DIEMS, Aurangabad

ABSTRACT

When we observe a sequencing of packet losses in the network where we are interesting to determine whether the losses are caused by link errors only. The particularly fascinated by the business executive attack case, whereby, malicious nodes that ar a part of the route exploit their data of communication context to by selection drop a little quantity of packets critical to network performance and we propose to exploit the correlations between lost packets. The packets dropping rates during this is corresponding to the channel error rate, standard algorithms is predicated on sleuthing the packets loss rate that can't succeed satisfactory detection accuracy. To reduce the computation overhead of the baseline schema, a packet -block-based mechanism is additionally planned, which allows one to trade detection accuracy for lower computation complexity. Furthermore, to ensure truthful calculation of these correlations, we develop a homomorphic linear authenticator (HLA) based public auditing architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes. This construction is privacy conserving, collusion proof, and incurs low communication and storage overheads. . Because the packet dropping rate during this case is corresponding to the channel error rate, standard algorithms that are supported sleuthing the packet loss rate cannot succeed satisfactory detection accuracy.

Keywords/ Index Term — Packet dropping, secure routing, attack detection, homomorphic linear signature, auditing

1. INTRODUCTION

In a multi-hop wireless network, nodes cooperate in relaying/routing traffic someone will exploit this cooperative nature to launch attacks. Eventually, such a severe denial-of-service (DoS) attack can paralyze the network by partitioning its topology. In the most severe kind, the malicious node simply stops forwarding every packet received from upstream nodes, completely disrupting the path between the source and the destination. For example, the adversary may first pretend to be a cooperative node in a route, the adversary starts dropping packets. First, the continuous presence of extremely high packet loss rate at the malicious nodes makes this type of attack easy to be detected [2]. Second, once being detected, these attacks are vulnerable. For example, in case the attack is detected but the malicious nodes are not identified, one can use the randomized multi-path routing algorithms [2], [3] to circumvent the black holes generated by the attack, probabilistically eliminating the attacker's threat. Even though persistent packet dropping will effectively degrade the performance of the network, from the attacker's point of view such an "always-on" attack has its disadvantages. If the malicious nodes are also identified, their threats can be completely eliminated by simply deleting these nodes from the network's routing table. A malicious node that's a part of the route will exploit its information of the network protocol and therefore the communication context to launch AN corporate executive attack—an attack that's irregular but can achieve the same performance degradation effect as a persistent attack at a much lower risk of being detected. Specifically, the malicious node might measure the importance of varied packets, so drop the tiny quantity that area unit deemed extremely essential to the operation of the network.

Sources Similarity In this paper, we are interested in combating such an insider attack. In particular, we are interested in the problem of detecting the occurrence of selective packet drops and identifying the malicious node(s) responsible for these drops. For example, in a frequency-hopping network, these could be the packets that convey frequency hopping sequences for network-wide frequency-hopping synchronization; in an ad hoc cognitive radio network, they could be the packets that carry the idle channel lists (i.e., white spaces) that are used to establish a network-wide control channel. By targeting these highly critical packets, the authors in [1], [4], [5] have shown that an intermittent insider attacker can cause significant damage to the network with low probability of being caught.

Specifically, due to the open nature of wireless medium, a packet drop in the network could be caused by harsh channel conditions (e.g., fading, noise, and interference, a.k.a., link errors), or by the insider attacker. Detecting selective packet-dropping attacks is extremely challenging in a highly dynamic wireless environment. The difficulty comes from the requirement that we need to not only detect the place (or hop) where the packet is dropped, but also identify whether the drop is intentional or unintentional. So, the insider attacker can camouflage under the background of harsh channel conditions. In this case, just by observing the packet loss rate is not enough to accurately identify the exact cause of a packet loss. In an open wireless environment, link errors are quite

significant, and may not be significantly smaller than the packet dropping rate of the insider attacker.

On the other hand, for the small number of works that differentiate between link errors and malicious packet drops, their detection algorithms usually require the number of maliciously-dropped packets to be significantly higher than link errors, in order to achieve an acceptable detection accuracy. The above problem has not been well addressed in the literature. As discussed in Section 2, most of the related works preclude the ambiguity of the environment by assuming that malicious dropping is the only source of packet loss, so that there is no need to account for the impact of link errors.

The high detection accuracy is achieved by exploiting the correlations between the positions of lost packets, as calculated from the auto-correlation function (ACF) of the packet-loss bitmap—a bitmap describing the lost/received status of each packet in a sequence of consecutive packet transmissions. The basic idea behind this method is that even though malicious dropping may result in a packet loss rate that is comparable to normal channel losses, the stochastic processes that characterize the two phenomena exhibit different correlation structures (equivalently, different patterns of packet losses).

In this paper, we develop an accurate algorithm for detecting selective packet drops made by insider attackers. Our algorithm also provides a truthful and publicly verifiable decision statistics as a proof to support the detection decision.

Therefore, by detecting the correlations between lost packets, one can decide whether the packet loss is purely due to regular link errors, or is a combined effect of link error and malicious drop. Our algorithm takes into account the cross-statistics between lost packets to make a more informative decision, and thus is in sharp contrast to the conventional methods that rely only on the distribution of the number of lost packets. The main challenge in our mechanism lies in how to guarantee that the packet-loss bitmaps reported by individual nodes along the route are truthful, i.e., reflect the actual status of each packet transmission.

This challenge is not trivial, because it is natural for an attacker to report false information to the detection algorithm to avoid being detected. For example, the malicious node may understate its packet-loss bitmap, i.e., some packets may have been dropped by the node but the node reports that these packets have been forwarded. Therefore, some auditing mechanism is needed to verify the truthfulness of the reported information. Considering that a typical wireless device is resource-constrained, we also require that a user should be able to delegate the burden of auditing and detection to some public server to save its own resources. The main challenge in our mechanism lies in how to guarantee that the packet-loss bitmaps reported by individual nodes along the route are truthful, i.e., reflect the actual status of each packet transmission. Such truthfulness is essential for correct calculation of the correlation between lost packets.

2. Literature Survey

The first class aims at high malicious dropping rates, wherever most (or all) lost packets area unit caused by malicious dropping. In this case, the impact of link errors is neglected. Most related work falls into this category. As a result, a maliciously node that continuous to drop packets will eventually deplete its credit, and will not be able to send its own traffic. The second sub-category is based on reputation systems. This name info is propagated sporadically throughout the network and is employed as a vital metric in choosing routes. Consequently, a malicious node will be excluded from any route. The third sub-category of works relies on end-to-end or hop-to-hop acknowledgements to directly locate the hops where packets are lost. Based on the methodology used to identify the attacking nodes, these works can be further classified into four subcategories. The first sub-category is based on credit systems [4], [8], [9]. A credit system provides an incentive for cooperation. A node receives credit by relaying packets for others, and uses its credit to send its own packets. Similarly, the method in [6], [3] traces the forwarding records of a particular packet at each intermediate node by formulating the tracing problem as a Renyi-Ulam game. The first hop where the packet is no longer forwarded is considered a suspect for misbehaving. A hop of high packet loss rate will be excluded from the route. The fourth subcategory addresses the problem using cryptographic methods. For example, the work in [7] utilizes Bloom filters to construct proofs for the forwarding of packets at each node. By examining the relayed packets at serial hops on a route, one can identify suspicious hops that exhibit high packet loss rates. All strategies mentioned on top of don't perform well once malicious packet dropping is extremely selective. More specifically, for the credit-system-based method, a malicious node may still receive enough credits by forwarding most of the packets it receives from upstream nodes. The works in [3] and [7] proposed to detect malicious packet dropping by counting the number of lost packets. If the quantity of lost packets is significantly larger than the expected packet loss rate created by link errors, then with high probability a malicious node is contributing to packet losses. Certain data of the wireless channel is critical during this case. The authors in [6] proposed to shape the traffic at the MAC layer of the source node according to a certain statistical distribution, so that intermediate nodes are able to estimate the rate of received traffic by sampling the packet arrival times. By comparing the source traffic rate with the estimated received rate, the detection algorithm decides whether the discrepancy in rates, if any, is within a reasonable range such that the difference can be considered as being caused by normal channel impairments only, or caused by malicious dropping, otherwise. Similarly, in the reputation-based approach, the malicious node can maintain a reasonably good reputation by forwarding most of the packets to the next hop. While the Bloom-filter theme is in a position to produce a packet forwarding proof, the correctness of the proof is probabilistic and it may contain errors. For highly selectively attacks (low packet-dropping rate), the intrinsic error rate of Bloom filter significantly undermines its detection accuracy. As for the acknowledgement-based technique and every one the mechanisms within the second class, simply numeration the quantity of lost packets doesn't provides a sufficient ground to sight the real culprit that is caused packet losses. The effort in the literature on this problem has been quite preliminary, and there is a few related works. Note that the cryptographic methods proposed in [4] to counter selective packet jamming target a different issue than the detection problem studied in this paper.

2.1 Network and Channel Models

We model the wireless channel of each hop along PSD as a random process that alternates between good and bad states. Packets transmitted during the good state are successful, and packets transmitted during the bad state are lost. In contrast to the classical Gilbert-Ellioit (GE) channel model, here we do not assume any Markovian property on the channel behavior. We only require that the sequence of sojourn times for each state follows a stationary distribution, and the autocorrelation

function of the channel state, say $f_s(i)$, where i is the time lag in packets, is also stationary. Here we limit our study to quasi-static networks, whereby the path PSD remains unchanged for a relatively long time, so that the link error statistics of the wireless channel is a wide-sense stationary (WSS) random process (i.e., $f_s(i)$ is stationary). Detecting malicious packet drops may not be a concern for highly mobile networks, because the fast-changing topology of such networks makes route disturbance is the main cause for packet losses. In this case, maintaining stable connectivity between nodes is a greater point than detecting malicious nodes. The function $f_s(i)$ can be calculated using the probing approach in [1]. In brief, a sequence of M packets are transmitted consecutively over the channel. By observing whether the transmissions are successful or not, the receiver obtains a realization of the channel $(a_{1,\dots,}a_M), where\ a_j \in \{0,1\} for\ j = 1,,,,,,,M$. In this sequence, "1" denotes the packet was successfully received, and "0" denotes the packet was dropped. $f_s(i)$ is derived by computing the autocorrelation function of this sample sequence: $f_s(i)^{def} = E\{a_j, a_{j+1}\} for\ i = 0,,,,,M$, where the expectation is calculated over all transmitted packets $j = 1,,,,,,,M$. This autocorrelation function describes the correlation between packet transmissions (successful/lost) at different times, as a function of the time lag. The time invariant nature of f_c is guaranteed by the WSS assumption of the wireless channel. The measurement of $f_c(i)$ can take place online or offline. A detailed discussion on how $f_c(i)$ is derived is out of the scope of this paper, and we simply assume that this information is given as input to our detection algorithm. Once being notified of possible attacks, S submits an attack-detection request (ADR) to Ad. To facilitate its investigation, Ad needs to collect certain information (elaborated on in the next section) from the nodes on route PSD. We assume that each such node must reply to Ad's inquiry, otherwise the node will be considered as misbehaving. We assume that normal nodes will reply with truthful information, but malicious nodes may cheat. At the same time, for privacy reasons, we require that Ad cannot determine the content of the normal packets delivered over PSD from the information collected during the auditing.

3. Proposed System

Under different packet dropping conditions, i.e., link-error versus malicious dropping, the instantiations of the packet-loss random process should present distinct dropping patterns (represented by the correlation of the instance). This is true even when the packet loss rate is similar in each instantiation. To verify this property, in Fig. 2 we have simulated the autocorrelation functions of two packet loss processes, one caused by 10 percent link errors, and the other by 10 percent link errors plus 10 percent malicious uniformly-random packet dropping. It can be observed that significant gap exists between these two auto-correlation functions. Therefore, by comparing the auto-correlation function of the observed packet loss process with that of a normal wireless channel (i.e., f_c(i), one can accurately identify the cause of the packet drops. The benefit of exploiting the correlation of lost packets can be better illustrated by examining the insufficiency of the conventional method that relies only on the distribution of the number of lost packets. More specifically, under the conventional method, malicious-node detection is modeled as a binary hypothesis test, where H0 is the hypothesis that there is no malicious node in a given link (all packet losses are due to link errors) and H1 denotes there is a malicious node in the given link (packet losses are due to both link errors and malicious drops). Let z be the observed number of lost packets on the link during some interval t. Then,:

$$z = \begin{cases} x, & under\ Ho\ (no\ malicious\ nodes) \\ x+y, & under\ H1\ (there\ is\ a\ malicious\ node) \end{cases} \quad (1)$$

where x and y are the numbers of lost packets caused by link errors and by malicious drops, respectively. Both x and y are random variables. Let the probability density functions of z conditioned on H0 and on H1 be h0ðzÞ and h1ðzÞ, respectively, as shown in Fig. 3a. We are interested in the maximum-uncertainty scenario where the a priori probabilities are given by $P_r\{Ho\} = P_r\{h1\} = 0.5$ i.e., the auditor has no prior knowledge of the distributions of H0 and H1 to make any biased decision regarding the presence of malicious nodes. Let the false-alarm and miss detection probabilities be P_{fa} and P_{md}, respectively. The optimal decision strategy that minimizes the total detection error $P_{\text{def}} = 0.5\,(P_{fa} + P_{md})$ is the maximum-likelihood (ML) algorithm:

$$\begin{cases} if\ z \le z_{th}, & accept\ \ ho \\ otherwise, & accept\ h1 \end{cases} \quad (2)$$

To correctly calculate the correlation between lost packets, it is critical to enforce a truthful packet-loss bitmap report by each node. We use HLA cryptographic primitive for this purpose. The basic idea of our method is as follows. An HLA scheme allows the source, which has knowledge of the HLA secret key, to generate HLA signatures $s_{1,,,,,,,}s_m$ for M independent messages $r_{1,,,,,,,}r_m$, respectively. The source sends out the $r_i's$ and $s_i's$ along the route. The HLA signatures are made in such a way that they can be used as the basis to construct a valid HLA signature for any arbitrary linear combination of the messages, $\sum_{i=1}^{M} C_i\, r_{i,}$, without the use of the HLA secret key, where C_i's are randomly chosen coefficients. A valid HLA signature for $\sum_{i=1}^{M} C_i\, r_{i,}$ can be constructed by a node that does not have knowledge of the secret HLA key if and only if the node has full knowledge of $s_{1,,,,,,,}s_m$. So, if a node with no knowledge of the HLA secret key provides a valid signature for $\sum_{i=1}^{M} C_i\, r_{i,,}$ it implies that this node must have received all the signatures $s_{1,,,,,,,}s_m$. Our construction ensures that si and ri are sent together along the route, so that knowledge of $s_{1,,,,,,,}s_m$ also proves that the node must have received $r_{1,,,,,,,}r_m$.

The HLA mechanism is used to detect the packet loses efficiently. In this mechanism we are finding the HLA signatures to each and every node along with the file. The HLA is used to find size of the given file. The HLA measures the size of the file along with the bandwidth or a threshold value. It selects the shortest distance between the neighboring nodes. Whenever we lose the packets due to malicious node the auditor can detect the traffic patterns and can recover by sending it again. It will stop the malicious node entering into the network

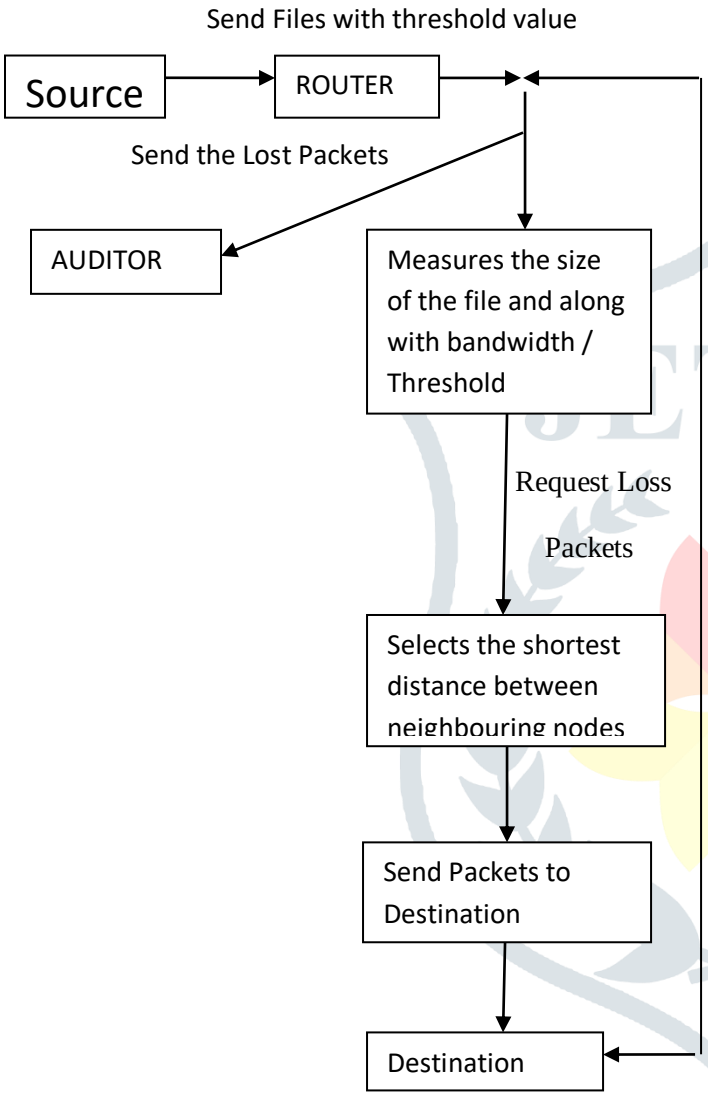

Fig 3.1 DFD Diagram To Detect packet Dropping Attacks

The public auditor Ad enters the detection phase after receiving and auditing the reply to its challenge from all nodes on P_{SD} . The main task so F_{AD} in this phase include the following: detecting any overstatement of packet loss at each node, constructing a packet-loss bitmap for each hop, calculating the autocorrelation function for the packet loss on each hop, and deciding whether malicious behavior is present. More specifically, Ad performs the set asks as follows. Given the packet-reception bitmap at each nod e$\overline{b_1}, \ldots, \overline{b_k}$, A_d, Ad first checks the consistency of the bitmaps for any possible overstatement of packet losses. Clearly, if there is no overstatement of packet loss, then the set of packets received at node j + 1 should be a subset of the packets received at node j, for j = 1, ,,,,,,, K – 1. Because a normal node always truthfully reports its packet reception, the packet-reception bitmap of a malicious node that overstates its packet loss must contradict with the bitmap of a normal downstream node. Note that there is always at least one normal downstream node, i.e., the destination D. So Ad only needs to sequentially scan $\overline{b_j}'s$ and the report from D to identify nodes that are overstating their packet losses.

For example, this can be done by combining the neighbor overhearing techniques [12] used in the reputation system. By fusing the testimony from the neighbors of these two nodes, Ad can pin-point the specific node that dropped the packet. Once being detected, the malicious node will be marked and excluded from the route to mitigate its damage

Although the optimal error threshold that minimizes the detection error is still an open problem, our simulations show that through trial-and-error, one can easily find a good th that provides a better detection accuracy than the optimal detection scheme that utilizes only the pdf of the number of lost packets. The above detection process applies to one end-to-end path. The detection for multiple paths can be performed as multiple independent detections, one for each path Furthermore, the privacy-preserving property of the ensures that publishing the auditing information will not compromise the confidentiality of the communication.

We consider the public auditor as a dedicated service provider that is not constrained by its computing capacity. So the computational overhead should not be a factor limiting the application of the algorithm at the public auditor. On the other hand, the proposed algorithm requires the source node to generate K HLA signatures for a K-hop path for each data packet. The generation of HLA signatures is computationally expensive, and may limit the applicability of the algorithm. We propose a block-based HLA signature and detection mechanism in Section 3, whereby the processing is based on block of packets rather than individual packets, to reduce this computation overhead by multiple folds. We evaluate the performance of the proposed mechanism by extensive simulations in incurred when PSD is established. Here we mainly focus on the recurring cost during the packet transmission and auditing phases (there is no communication overhead in the detection phase). For a transmitted packet Pi, S needs to send one encrypted HLA signature and one MAC to each intermediate node on PSD. Our HLA signature follows the BLS scheme in [7]. So an HLA signature sij is 160-bit long. If encrypted by DES, the encrypted signature ~ sij is 192 bits in length (a block in DES is 64-bit long, so the length of the cipher text of DES is multiples of 64 bits). The MAC-related hash function HMAC key can be implemented in SHA-1 and has a length of 160 bits. So for each packet, the per-hop communication overhead incurred by the proposed scheme in the packet transmission phase is 192 + 160 = 352 bits, or 44 bytes. For a path of K intermediate hops, the total communication overhead for transmitting a packet is 44K bytes. For example, when K = 10, the overhead is 440 bytes/ packet. For an IEEE 802.11 system, this is about 19 percent of the maximum MSDU. HLA detection algorithm is the high computation overhead of the source node. In this section, we proposed a block-based solution that can reduce this overhead by multiple folds. The main idea is to make the HLA signature scalable: instead of generating per-packet HLA signatures, per-block HLA signatures will be generated, where a block consists of L>1 packets. Accordingly, the detection will be

extended to blocks, and each bit in the packet-loss bitmap represents a block of packets rather than a single packet. The details of this extension are elaborated as follows. Auditing is now based on blocks. In particular, at node nj, suppose the sequence number of the blocks recorded in the proof-of-reception database are B1, ….., BM. Based on the information in the database, node generates a block reception bitmap , where bji = 1 if and only if all L packets in block Bi has been received by , and bji = 0 otherwise. Except the above, Ad still follows the same algorithm in Section 4.2.3 to submit random challenge, receive response, and verify the truthfulness of the reported block-reception bitmap. In the detection phase, the ACF of the wireless channel needs to be coarsened such that one unit of lag represents L consecutive packets. This could be done by first coarsening the packet reception bitmap observed in the training phase using blocks: L consecutive 1's are mapped to a 1 in the blocked-based bitmap, otherwise a 0 will be mapped. The ACF of the coarsened wireless channel is then compared with the ACF of the block-reception bitmap reported by each node to detect possible malicious packet drops. From the above description, it is clear that the block based HLA signature and detection mechanism can in general reduce the computation overhead by L folds. However, the coarser representation of lost packets makes it difficult to accurately capture the correlation between them. For example, even with a small block size, say L = 2, it is not possible to tell whether a block loss is due to the loss of one packet or both packets in the block, which correspond to very different packet-loss correlations. Therefore, it is expected that the reduced computational overhead comes at the cost of less detection accuracy.

4. Conclusion

So this Paper is compared with many different algorithms where only they utilize distributions of the no of lost packets, which improves the detection of the malicious packet loss which are loss significantly. So based on HLA based algorithm which is auditing packet loss reporting by individual nodes . Source and destination is pursued because of delivering packets end to end. Some open issues is to be explored which are limited to static or quasi static wireless adhoc network and finally packets rates are under two following rates i.e high and low so this property can be computed complexity for detection accuracy which is universal..

REFERENCES

[1] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security, 2009, pp. 319–333.

[2] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks," ACM Trans. Inf. Syst. Secur., vol. 10, no. 4, pp. 11–35, 2008.

[3] L. Buttyan and J. P. Hubaux, "Stimulating cooperation in selforganizing mobile ad hoc networks," ACM/Kluwer Mobile Netw. Appl., vol. 8, no. 5, pp. 579–592, Oct. 2003.

[4] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputation-based incentive scheme for ad hoc networks," in Proc. IEEE Wireless Commun. Netw. Conf., 2004, pp. 825–830.

[5] P. Papadimitratos and Z. Haas, "Secure message transmission in mobile ad hoc networks," Ad Hoc Netw., vol. 1, no. 1, pp. 193–209, 2003.

[6] T. Shu, S. Liu, and M. Krunz, "Secure data collection in wireless sensor networks using randomized dispersive routes," in Proc. IEEE INFOCOM Conf., 2009, pp. 2846–2850.

[7] R. Rao and G. Kesidis, "Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," in Proc. IEEE GLOBECOM Conf., 2003, pp. 2957–2961.

[8] Y. Xue and K. Nahrstedt, "Providing fault-tolerant ad-hoc routing service in adversarial environments," Wireless Pers. Commun., Special Issue Secur. Next Generation Commun., vol. 29, no. 3, pp. 367– 388, 2004.

[9] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple cheat-proof, credit-based system for mobile ad-hoc networks," in Proc. IEEE INFOCOM Conf., 2003, pp. 1987–1997.