ISSN No: 2454-9614

Privacy-Couple Based Detection of Compromised Nodes of Packet Dropping Attacks in Wireless AD-HOC Networks

V. Gayathri, V. Meena*, R. Gunasekaran, C. Vasuki

Department of InformartionTechnology, Nandha Engineering College,
Perundura, Erode-52, Tamilnadu, India.

*Corresponding Author: V.Gayathri

E-mail: krihnakumarbtech@gmail.com

Received: 12/11/2015, Revised: 14/12/2015 and Accepted: 5/03/2016

Abstract:

Link error and malicious packet dropping are two sources for packet losses in multi-hop wireless ad hoc network. In this paper, while observing a sequence of packet losses in the network, we are interested in determining whether the losses are caused by link errors only, or by the combined effect of link errors and malicious drop. We are especially interested in the insider-attack case, whereby malicious nodes that are part of the route exploit their knowledge of the communication context to selectively drop a small amount of packets critical to the network performance. Because the packet dropping rate in this case is comparable to the channel error rate, conventional algorithms that are based on detecting the packet loss rate cannot achieve satisfactory detection accuracy. To improve the detection accuracy, we propose to exploit the correlations between lost packets. 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 preserving, collusion proof, and incurs low communication and storage overheads. To reduce the computation overhead of the baseline scheme, a packet-block-based mechanism is also proposed, which allows one to trade detection accuracy for lower computation complexity. Through extensive simulations, we verify that the proposed mechanisms achieve significantly better detection accuracy than conventional methods such as a maximum-like.

Reviewed by **ICETSET'16** *organizing committee*

1. Introduction

In like hood based detection, a multi-hop wireless network, nodes cooperate in relaying/routing traffic. An adversary can exploit this cooperative nature to launch attacks. For example, the adversary may first pretend to be a cooperative node in the route discovery process. Once being included in a route, the adversary starts dropping packets. In the most severe form, the malicious node simply stops forwarding every packet received from upstream nodes, completely disrupting the path between the source and the destination. Eventually, such a severe denial-of-service (DoS) attack can paralyze the network by partitioning its topology. Even though persistent packet dropping can effectively degrade the performance of the network; from the attacker's standpoint such an "always-on" attack has its disadvantages. First, the continuous presence of extremely high packet loss rate at the malicious nodes makes this type of attack easy to be detected **[25].** Second, once being detected, these attacks are easy to mitigate. For example, in case the attack is detected but the malicious nodes are not identified, one can use the randomized multi-path routing algorithms **[28], [29]** to circumvent the black holes generated by the attack, probabilistically eliminating the attacker's threat. If the malicious nodes are also identified, their threats can be completely eliminated by simply deleting these nodes from the network's routing table.

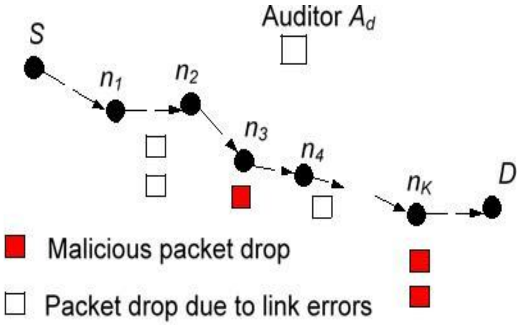
A malicious node that is part of the route can exploit its knowledge of the network protocol and the communication context to launch an insider attack an attack that is intermittent, but can achieve the same performance degradation effect as a persistent attack at a much lower risk of being detected. Specifically, the malicious node may evaluate the importance of various packets, and then drop the small amounts that are deemed highly critical to the operation of the network. 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 **[21], [24], [25]** have shown that an intermittent insider attacker can cause significant damage to the network with low probability of being caught. 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.

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. 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.

2. Related work

Depending on how much weight a detection algorithm gives to link errors relative to malicious packet drops, the related work can be classified into the following two categories. The first category aims at high malicious dropping rates, where most (or all) lost packets are caused by malicious dropping. In this case, the impact of link errors is ignored. Most related work falls into this category. Based on the methodology used to identify the attacking nodes, these works can be further classified into four sub-categories. The first sub-category is based on credit systems **[9], [34], [10].** 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. 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 **[12], [8], [14], [19], [20], [11], [4].** A reputation system relies on neighbours to monitor and identify misbehaving nodes. A node with a high packet dropping rate is given a bad reputation by its neighbours. This reputation information is propagated periodically throughout the network and is used as an important metric in selecting 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 **[18], [22], [23], [5], [6], [32].**

A hop of high packet loss rate will be excluded from the route. The fourth sub-category addresses the problem using cryptographic methods. For example, the work in **[17]** utilizes Bloom filters to construct proofs for the forwarding of packets at each node. By examining the relayed packets at successive hops along a route, one can identify suspicious hops that exhibit high packet loss rates. Similarly, the method in **[16], [33]** traces the forwarding records of a particular packet at each inter-mediate 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. The second category targets the scenario where the number of maliciously dropped packets is significantly higher than that caused by link errors, but the impact of link errors is non-negligible. Certain knowledge of the wireless channel is necessary in this case. The authors in **[26]** proposed to shape the traffic at the MAC layer of the source node according to a certain statistical distribution, so that inter-mediate nodes are able to estimate the rate of received traffic by sampling the packet arrival times.



3. System Models and Problem Statement

3.1 Network and Channel Models

Consider an arbitrary path P_{SD} in a multi-hop wireless ad hoc network, as shown in Fig. 1. The source node S continuously sends packets to the destination node D through inter-mediate nodes n_1;. . .; n_K , where n_i is the upstream node of $n_{iþ1}$, for 1 i K 1. We assume that S is aware of the route P_{SD}, as in Dynamic Source Routing (DSR) **[15].** If DSR is not used, S can identify the nodes in P_{SD} by performing a trace route operation. Here we mainly focus on static or quasi-static wireless ad hoc networks, i.e., we assume that the network topology and link characteristics remain unchanged for a relatively long period of time. Example net-works include wireless mesh networks (WMNs) and ad hoc networks formed in nomadic computing. Extension to a highly mobile environment is out of our scope and will be considered in the future work.

We model the wireless channel of each hop along P_{SD} 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 behaviour. 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 $f_c$ðiÞ, where i am the time lag in packets, is also stationary. Here we limit our study to quasi-static networks, whereby the path P_{SD} 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., $f_c$ðiÞ 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 disruption the dominant cause for packet losses. In this case, maintaining stable connectivity between nodes is a greater concern than detecting malicious nodes. The function $f_c$ðiÞ can be calculated using the probing approach in [1]. In brief, sequences of M packets are transmitted consecutively over the channel. By observing whether the trans-missions are successful or not, the receiver obtains a realization of the channel state ða_1;. . .; a_MÞ, where a_j 2 f0; 1g for j ¼ 1; . . . ; M. In this sequence, “1” denotes the packet was successfully received, and “0” denotes the packet was dropped. $f_c$ðiÞ is derived by computing the autocorrelation function of this sample sequence:

4. Proposed Detection Scheme

4.1 Overview

The proposed mechanism is based on detecting the correlations between the lost packets over each hop of the path. The basic idea is to model the packet loss process of a hop as a random process alternating between 0 (loss) and 1 (no loss). Specifically, consider that a sequence of M packets that are transmitted consecutively over a wireless channel. By observing whether the transmissions are successful or not, the receiver of the hop obtains a bitmap ða_1;. . .; a_MÞ, where a_j2 f0;1g for packets j¼1;. . .; M. The correlation of the lost packet is calculated as the auto-correlation function of this bitmap. 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 auto-correlation 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

4.2 Scheme Details

4.2.1 Setup Phase

This phase takes place right after route P$_{SD}$ is established, but before any data packets are transmitted over the route. In this phase, S decides on a symmetric-key crypto-system ðencrypt$_{key}$; decrypt$_{key}$Þand K symmetric keys key$_1$; . . . ; key$_K$, where encrypt$_{key}$ and decrypt$_{key}$ are the keyed encryption and decryption functions, respectively. S securely dis-tributes decrypt$_{key}$ and a symmetric key key$_j$ to node n$_j$ on P$_{SD}$, for j ¼ 1; . . . ; K. Key distribution may be based on the public-key crypto-system such as RSA: S encrypts key$_j$ using the public key of node n$_j$ and sends the cipher text to n$_j$.n$_j$ decrypts the cipher text using its private key to obtainkey$_j$. S also announces two hash functions, H$_1$andH$_{key}^{MAC}$, to all nodes in P$_{SD}$. H$_1$ is un-keyed while H$_{key}^{MAC}$ is a keyed hash function that will be used for message authentication purposes later on.

4.2.3 Audit Phase

This phase is triggered when the public auditor A$_d$ receives an ADR message from S. The ADR message includes the id of the nodes on P$_{SD}$, ordered in the downstream direction, i.e., n$_1$;. . .; n$_K$,S’s HLA public key information pk¼ðv; g; uÞ, the sequence numbers of the most recent M packets sent by S, and the sequence numbers of the subset of these M packets that were received by D. Recall that we assume the information sent by S and D is truthful, because detecting attacks is in their interest. A$_d$ conducts the auditing pro-cess as follows.

A$_d$ submits a random challenge vector ~c$_j$ ¼ ðc$_{j1}$; . . . ; c$_{jM}$ Þ to node n$_j$, j¼1;. . .; K, where the elements c$_{ji}$’s are randomly chosen from Z$_p$. Without loss of generality, let the sequence number of the packets recorded in the current proof-of-reception database be P$_1$;. . .; P$_M$, with P$_M$ being the most recent packet sent by S. Based on the information in this database, node n$_j$ generates a packet-reception bit-

4.3 Overhead Analysis

The proposed scheme requires relatively high computation capability at the source, but incurs low communication and storage overheads along the route, as explained below.

4.3.1 Computation Requirements

Most of the computation is done at the source node (for generating HLA signatures) and at the public auditor (for con-ducting the detection process). We consider the public auditor as a dedicated service provider that is not con-strained 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 5, 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.

4.3.2 Communication Overhead

The communication overhead for the setup phase is a one-time cost, incurred when P_{SD} 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 P_i, S needs to send one encrypted HLA signature and one MAC to each intermediate node on P_{SD}. Our HLA signature follows the BLS scheme in **[7].** So an HLA signature s_{ij} is 160-bit long. If encrypted by DES, the encrypted signature $s\sim_{ij}$ 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 H_{key}^{MAC} 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 192þ160¼352 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 K¼10, the overhead is 440 bytes/ packet. For an IEEE 802.11 system, this is about 19 percent of the maximum MSDU (2,304 bytes).

In the auditing phase, the auditor A_d sends a random challenge vector $\sim c_j$ to each node n_j. Let each element in this vector be a 32-bit integer. The challenge has a length of 4M bytes. Based on our simulation in Section 6, M¼50 is typically enough to achieve good detection accuracy. So this means each challenge can be delivered in one packet. M- bit map. $r^{\text{ðjÞ}}$is the linear combination of the SHA-1image of the packets, so $r^{\text{ðjÞ}}$ also has a length of 160 bits. $s^{\text{ðjÞ}}$ is an HLA signature of $r^{\text{ðjÞ}}$, so it is also 160-bit long. Overall, the reply from a node to A_d has a length of 320þM bits, which can also be delivered in one packet.

4.3.3 Storage Overhead

During its operation, a node n_j on P_{SD} needs to store the key key_j, the H_1 hash image, and the associated HLA sig-nature for each of the M most recently received packets. Assuming $encrypt_{key}$ and $decrypt_{key}$ are based on DES, key_j has a length of 56 bits. Let the hash function H_1 be based on SHA-1. So the H_1 image of a packet is 160-bit long. The HLA signature is based on BLS (Boneh-Lynn-Shacham) scheme **[7]** and is 160-bit long. So in total the storage over-head at n_j is 320Mþ56 bits, or 40Mþ7 bytes. This storage overhead is quite low. For example, when M50, the storage overhead at a node is less than 2 KB.

5. Reducing Computation Overhead:

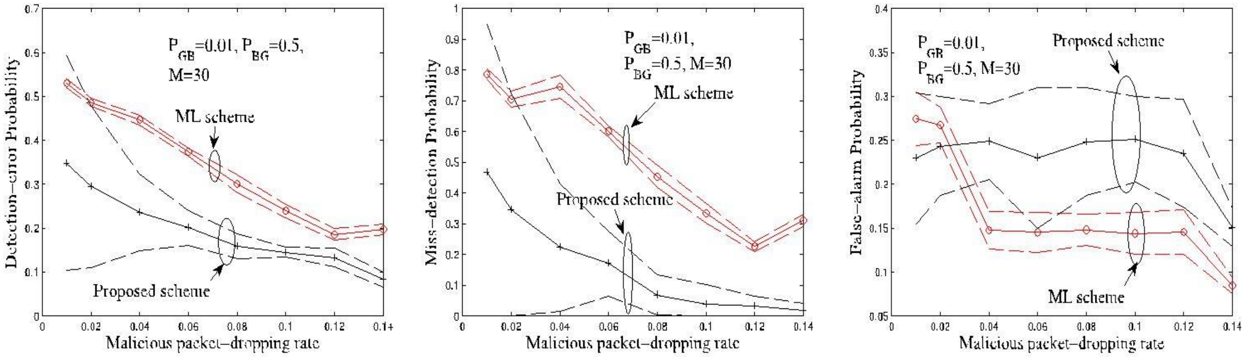
5.1 Block-Based HLA Signature Generation and Detection

One major limitation of the proposed baseline 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 packet. 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.

In the Packet Transmission Phase, rather than generating HLA signatures for every packet, now the signatures are based on a block of packets. In particular, L consecutive packets are deemed as one block. Accordingly, the stream of packets is now considered as stream of blocks. Denote the L packets in block i as P_{i1};. . .; P_{iL}, respectively.

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 L¼2, 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.



(a) Overall detection-error probability (b) Miss-detection probability (c) False-alarm probability

6. Results

6.1 Random Packet Dropping

The detection accuracy is shown in Fig. 4 as a function of the malicious random-drop rate P_M . In each subfigure, there are two sets of curves, representing the proposed algorithm and the optimal ML scheme, respectively. In each set of curves, the one in the middle represents the mean, and the other two represent the 95 percent confidence interval. In general, the detection accuracy of both algorithms improves with P_M (i.e., the detection error decreases with P_M). This is not surprising, because malicious packet drops become more statistically distinguishable as the attacker starts to drop more packets. In addition, this figure shows that for $_{th}$¼10%, the proposed algorithm provides slightly higher false-alarm rate (subfigure (c)) but significantly lower miss-detection probability (subfigure (b)) than the ML scheme. A low miss-detection probability is very desirable in our context, because it means a malicious node can be detected with a higher probability. The slightly higher false-alarm rate should not be a problem, because a false alarm can be easily recognized and fixed in the post-detection investigation phase. Most importantly, the overall detection-error probability of the proposed scheme is lower than that of the ML scheme (subfigure (a)).

In which the attacker hides its drops in the background of link errors by mimicking the channel-related loss rate. In this case, the ML scheme cannot correctly differentiate between link errors and malicious drops. For example, when P_M¼0:01, the ML scheme results in P_{md}¼80% and P_{fa}¼23%. This is close to arbitrarily ruling that every packet loss is due to link error only, leading to an overall detection-error rate of 50 per-cent (see subfigure (a)). Our proposed algorithm, on the other hand, achieves much better detection accuracy, because it's P_{md} and $P_{fa}$are both lower than those under the ML scheme.

As a result, when P_M¼0:01, the total detection-error rate of the proposed algorithm is about 35 percent. When P_M is increased to 0.04, P_{error} of the proposed scheme reduces to only 20 percent, which is roughly half of the error rate of the ML scheme at the same P_M . Remembering that the detection-error rate of the ML scheme is the lowest among all detection schemes that only utilize the distribution of the number of lost packets, the lower detection-error rate of the proposed scheme shows that exploiting the correlation between lost packets helps in identifying the real cause of packet drops more accurately. The effect of exploiting the correlation is especially visible when the malicious packet-drop rate is com-parable with the link error rate. Meanwhile, we also note that the 95 percent confidence interval of the proposed scheme is wider than that of the ML scheme. This is because the decision variable ${}_j$ in the proposed scheme is a second-order function of the random packet loss process, while the decision variable in the ML scheme (i.e., number of lost packets) is a first order function of the same packet loss process. As a result, the decision variable of the proposed scheme possesses more randomness than that of the ML scheme, as reflected by the wider 95 percent confidence interval.

6.1.1 Selective Packet Dropping

The detection error as a function of the number of maliciously dropped packets is shown in Fig. 7. At the low end of the x-axis, maliciously dropped packets account for only 1=50 ¼ 2% of the total packets in the packet-loss bitmap. This is identical to the link error rate of 0.02, assumed in the simulation. Similar performance trends can be observed to the case of the random packet dropping. Fewer detection errors are made by both algorithms when more packets are maliciously dropped. In all the simulated cases, the pro-posed algorithm can detect the actual cause of the packet drop more accurately than the ML scheme, especially when the number of maliciously dropped packets is small. When the number of maliciously dropped packets is significantly higher than that caused by link errors (greater than four packets in our simulation), the two algorithms achieve com-parable detection accuracy. In this scenario, it may be wise to use the conventional ML scheme due to its simplicity (e.g., no need to enforce truthful reports from intermediate nodes, etc.).

The detection errors are plotted in Fig. 8 as a function of the size of the packet-loss bitmap (M). To conduct a fair comparison, as we increase M, we also increase the number of maliciously dropped packets, so as to maintain a malicious packet-dropping rate of 10 percent. It can be observed that a small M is enough to achieve good detection accuracy.

6.1.2 Dropping of Control Packets

Our simulations so far have not made any application-semantic (use case) assumption on the dropped packets. In reality, however, because these packets are usually used for control purposes, the loss of these packets may generate significant impacts on the transmission of other (i.e., data) packets. In this series of simulations, we evaluate how the correlation between the control and data packets affects the performance of the proposed scheme. In particular, we con-sider a multi-hop cognitive radio network, where control packets are exchanged over an end-to-end path to maintain channel synchronization between consecutive hops. A control packet contains the channel id that the two ends of a hop should tune to. The exchange of these packets could be end-to-end (i.e., the entire path operates over a channel commonly available to all hops) or local (the path consists of several segments, each of which operates over a different channel available only to the hops of that segment). In either case, the drop of a control packet will disrupt the frequency.

6.1.3 Block-Based Detection

In this series of simulations, we study the detection accuracy of block-based algorithms as a function of block size. Fig. 11a plots the detection accuracy for random packet drops under two packet drop probabilities: high (P_M¼0:08) and low (P_M¼0:01). The performance of the ML scheme is also plotted in the same figure for comparison. In general, it shows that for both cases the detection error increases with the block size. This is expected, as a larger block size hides more details of packet losses, and therefore makes the actual correlation of lost packets more difficult to calculate. Mean-while, the benefits of blocked-based algorithm are also observed: it is able to trade computation complexity for better detection accuracy. For example, under low packet drop-ping rate, it shows that the block-based algorithm can reduce the computation overhead of the baseline HLA detection by 10 folds, and still be able to achieve better detection accuracy than the ML scheme. At high packet dropping rate, the block-based algorithm can achieves a 4computa-tion overhead reduction, while still achieving slightly better detection accuracy than the ML scheme.

7. Conclusions

In this paper, we showed that compared with conventional detection algorithms that utilize only the distribution of the number of lost packets, exploiting the correlation between lost packets significantly improves the accuracy in detecting malicious packet drops. Such improvement is especially visible when the number of maliciously dropped packets is comparable with those caused by link errors. To correctly calculate the correlation between lost packets, it is critical to acquire truthful packet-loss information at individual nodes. We developed an HLA-based public auditing architecture that ensures truthful packet-loss reporting by individual nodes. This architecture is collusion proof, requires relatively high computational capacity at the source node, but incurs low communication and storage overheads over the route. To reduce the computation overhead of the baseline construction, a packet-block-based mechanism was also proposed, which allows one to trade detection accuracy for lower computation complexity.

Some open issues remain to be explored in our future work. First, the proposed mechanisms are limited to static or quasi-static wireless ad hoc networks. Frequent changes on topology and link characteristics have not been considered. Extension to highly mobile environment will be studied in our future work. In addition, in this paper we have assumed that source and destination are truthful in following the established protocol because delivering packets end-to-end is in their interest. Misbehaving source and destination will be pursued in our future research. Moreover, in this paper, as a proof of concept, we mainly focused on showing the feasibility of the proposed crypto-primitives and how second-order statistics of packet loss can be utilized to improve detection accuracy. As a first step in this direction, our analysis mainly emphasize the fundamental features of the problem, such as the untruthfulness nature of the attackers, the public verifiability of proofs, the privacy-preserving requirement for the auditing process, and the randomness of wire-less channels and packet losses, but ignore the particular behaviour of various protocols that may be used at different layers of the protocol stack. The implementation and optimization of the proposed mechanism under various particular protocols will be considered in our future studies.

References:

[1] J. N. Arauz, "802.11 Markov channel modeling," Ph.D. dissertation, School Inform. Sci., Univ. Pittsburgh, Pittsburgh, PA, USA, 2004.

[2] C. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. ACM Conf. Comput. andCommun. Secur., Oct. 2007, pp. 598–610.

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

[4] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure byzantine resilient rout-ing protocol for wireless ad hoc networks," ACM Trans. Inform. Syst. Security, vol.

[5] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. [28] T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless Rubens, "ODSBR: An on-demand secure byzantine resilient rout-ing protocol for wireless ad hoc networks," ACM Trans. Inf. Syst. Secur., vol. 10, no. 4, pp. 11–35, 2008.

[6] K. Balakrishnan, J. Deng, and P. K. Varshney, "TWOACK: Pre-venting selfishness in mobile ad hoc networks," in Proc. IEEE Wireless Commun. Netw. Conf., 2005, pp. 2137–2142.

[7] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," J. Cryptol., vol. 17, no. 4, pp. 297–319, Sep. 2004.

[8] S. Buchegger and J. Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)," in Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput. Conf., 2002, pp. 226–236.

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

[10] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring, "Modelling incentives for collaboration in mobile ad hoc networks," pre-sented at the First Workshop Modeling Optimization Mobile, Ad Hoc Wireless Netw., Sophia Antipolis, France, 2003.

[11] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Routing amid colluding attackers," in Proc. IEEE Int. Conf. Netw. Protocols, 2007, 184–193.

[12] W. Galuba, P. Papadimitratos, M. Poturalski, K. Aberer, Z. Despotovic, and W. Kellerer, "Castor: Scalable secure routing for ad hoc networks," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1 –9.

[13] T. Hayajneh, P. Krishnamurthy, D. Tipper, and T. Kim, "Detecting malicious packet dropping in the presence of collisions and chan-nel errors in wireless ad hoc networks," in Proc. IEEE Int. Conf. Commun., 2009, pp. 1062–1067.

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

[15] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in Ad Hoc Networking. Reading, MA, USA: Addison-Wesley, 2001, ch. 5, pp. 139–172.

[16] W. Kozma Jr. and L. Lazos, "Dealing with liars: Misbehavior identification via Renyi-Ulam games," presented at the Int. ICST Conf. Security Privacy in Commun. Networks, Athens, Greece, 2009.

[17] W. Kozma Jr., and L. Lazos, "REAct: Resource-efficient accountability for node misbehavior in ad hoc networks based on random audits," in Proc. ACM Conf. Wireless Netw. Secur., 2009, pp. 103–110.

[18] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan, "An acknowl-edgement-based approach for the detection of routing misbehav-ior in MANETs," IEEE Trans. Mobile Comput., vol. 6, no. 5, 536–550, May 2006.

[19] Y. Liu and Y. R. Yang, "Reputation propagation and agreement in mobile ad-hoc networks," in Proc. IEEE WCNC Conf., 2003, 1510–1515.

[20] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing mis-behavior in mobile ad hoc networks," in Proc. ACM MobiCom Conf., 2000, pp. 255–265.

[21] G. Noubir and G. Lin, "Low-power DoS attacks in data wireleslans and countermeasures," ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 7, no. 3, pp. 29–30, Jul. 2003.

[22] V. N. Padmanabhan and D. R. Simon, "Secure traceroute to detect faulty or malicious routing," in Proc. ACM SIGCOMM Conf., 2003,77–82.

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

[24] A. Proano and L. Lazos, "Selective jamming attacks in wireless networks," in Proc. IEEE ICC Conf., 2010, pp. 1–6.

[25] A. Proano and L. Lazos, "Packet-hiding methods for preventing selective jamming attacks," IEEE Trans. Depend. Secure Comput., vol. 9, no. 1, pp. 101–114, Jan./Feb. 2012.