ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

Secure way of Truthful Detection of Packet Dropping Attack in Wireless AdHoc Networks

Revathi.M[1], Subbulakshmi.G[2]

PG Scholar, Department of CSE, Vandayar Engineering College, Thanjavur, India [1]

Assistant Professor, Department of CSE, Vandayar Engineering College, Thanjavur, India [2]

***Abstract*—Link error and malicious packet dropping are two sources for packet losses in multi-hop wireless ad hoc network. While observing a sequence of packet losses in the network, whether the losses are caused by link errors only, or by the combined effect of link errors and malicious drop are to be identified. 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. Misbehaving source and destination will be detected by intrusion detection. To improve the detection accuracy, the correlations between lost packets is identified. Homo-morphic linear authenticator (HLA) based public auditing architecture is developed 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**

***Index Terms*— Packet dropping, secure routing, attack detection, homo-morphic linear signature, auditing, intrusion detection.**

I. INTRODUCTION

In 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 [12]. 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 [12], [13] 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., whitespaces) that are used to establish a network-wide control channel. By targeting these highly critical packets, the authors in [12] 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.

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. 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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

a packet loss. 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. 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 acceptable detection accuracy. 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. 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). 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., reflects the actual status of each packet transmission. Such truthfulness is essential for correct calculation of the correlation between lost packets. 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.

Our solution to the above public-auditing problem is constructed based on the homo-morphic linear authenticator(HLA) cryptographic primitive [1], [2], which is basically signature scheme widely used in cloud computing and storage server systems to provide a proof of storage from the server to entrusting clients . However, direct application of HLA does not solve our problem well, mainly because in our problem setup, there can be more than one malicious node along the route. These nodes may collude (by exchanging information) during the attack and when being asked to submit their reports. For example, a packet and its associated HLA signature may be dropped at an upstream malicious node, so a downstream malicious node does not receive this packet and the HLA signature from the route. However, this downstream attacker can still open a back-channel to request this information from the upstream malicious node. When being audited, the downstream malicious node can still provide valid proof for the reception of the packet. So packet dropping at the upstream malicious node is not detected. Such collusion is unique to our problem, because in the cloud computing/storage server scenario, a file is uniquely stored at a single server, so there are no other parties for the server to collude with. We show that our new HLA construction is collusion-proof. Our construction also provides the following new features.

First, privacy-preserving: the public auditor should not be able to discern the content of a packet delivered on the route through the auditing information submitted by individual hops, no matter how many independent reports of the auditing information are submitted to the auditor. Second, our construction incurs low communication and storage overheads at intermediate nodes. This makes our mechanism applicable to a wide range of wireless devices; including low-cost wireless sensors that have very limited bandwidth and memory capacities.

This is also in sharp contrast to the typical storage-server scenario, where bandwidth/storage is not considered an issue. Last, to significantly reduce the computation overhead of the baseline constructions so that they can be used in computation-constrained mobile devices, a packet-block-based algorithm is proposed to achieves scalable signature generation and detection. This mechanism allows one to trade detection accuracy for lower computation complexity.

Intrusion is defined as “any set of actions that attempt to compromise the integrity, confidentiality, or availability of a host or to notify another participating node for the malicious action of the node that performs the attack resource”. Intrusion protection techniques captures audit data and perform traffic analysis to detect whether the network or a specific node is under attack. The two types of nodes are is under attack on a network. First type selfish node doesn’t cooperate for selfish reasons, such as saving power. The main threat from selfish nodes is the dropping of packets, which may affect the performance of the network severely. Second type malicious node has the intention to damage other nodes, and battery saving is not a priority.

II. 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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

first sub-category is based on credit systems [9], [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], [10]. A reputation system relies on neighbors to monitor and identify misbehaving nodes. Anode with a high packet dropping rate is given a bad reputation by its neighbors. 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 [4], [5]. 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 [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], 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. 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 intermediate nodes are able to estimate the rate of received traffic by sampling the packet arrival times.

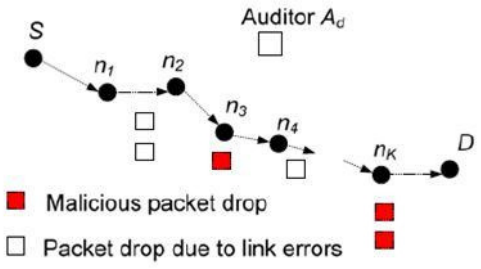


Fig 1 Network and Attack Model

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. The works in [9] and [13] proposed to detect malicious packet dropping by counting the number of lost packets. If the number of lost packets is significantly larger than the expected packet loss rate made by link errors, then with high probability a malicious node is contributing to packet losses.

All methods mentioned above do not perform well when malicious packet dropping is highly 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. 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 schemes able to provide 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 filer significantly undermines its detection accuracy. As for the acknowledgement-based method and all the mechanisms in the second category, merely counting the number of lost packets does not give a sufficient ground to detect the real culprit that is causing packet losses. This is because the difference in the number of lost packets between the link-error only case and the link-error-plus-malicious-dropping case is small when the attacker drops only a few packets. Consequently, the detection accuracy of these algorithms deteriorates when malicious drops become highly selective.

Our study targets the challenging situation where link errors and malicious dropping lead to comparable packet loss rates. 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 [11] to counter selective packet jamming target a different issue than the detection problem studied in this paper. The methods in [11] delay a jammer from recognizing the significance of a packet after the packet has been successfully transmitted, so that there is no time for the jammer to conduct jamming based on the content/importance of the packet.

Instead of trying to detect any malicious behavior, the approach in [11] is proactive, and hence incurs overheads regardless of the presence or absence of attackers.

III. Proposed Detection Scheme

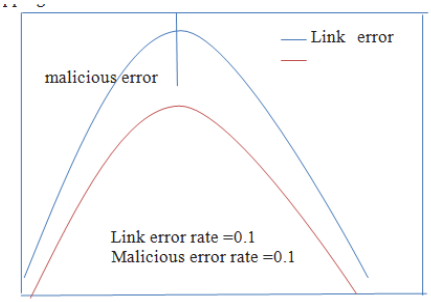
The proposed mechanism is based on detecting the correlations between the lost packets over each hop of the path. The basic idea is to model the packet loss process of a hop as a random process alternating between 0 (loss) and 1 (no loss). Specifically, consider that a sequence of M packets that are transmitted consecutively over a wireless channel. By observing whether the transmissions are successful or not, the receiver of the hop obtains a bitmap (a1; . . . ; aM), where aj€{0; 1} for packets j = 1; . . .;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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

link errors, and the other by 10 percent link errors plus10 percent malicious uniformly-random packet dropping.

The SI unit for magnetic field strength *H* is A/m. However, if you wish to use units of T, either refers to magnetic flux density *B* or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."



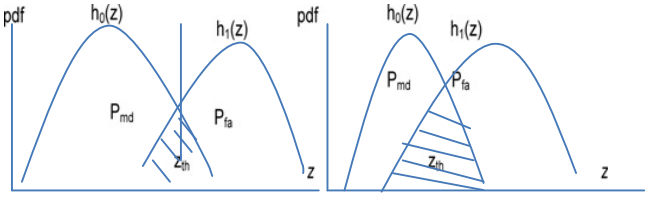
It can be observed that significant gap exists between these two auto-correlation functions. Therefore, by comparing the auto-correlation function of the observed packet loss process with that of a normal wireless channel (i.e., fc(i)), one can accurately identify the cause of the packet drops. The benefit 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,

$$Z = \begin{cases} x; & \text{under } H_0 \text{ (no malicious node)} \\ x + y; & \text{under } H_1 \text{ (there is a malicious node);} \end{cases} \quad -(1)$$

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 H_0 and on H_1 be $h_0(z)$ and $h_1(z)$, respectively, as shown in Fig. 3a. We are interested in the maximum-uncertainty scenario where the prior probabilities are given by Pr $\{H_0\}$ =Pr $\{H_1\}$ = 0:5, i.e., the auditor has no prior knowledge of the distributions of $H_0$and H_1 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_{de}def=0:5(P_{fa}+P_{md}) is the maximum-likelihood (ML) algorithm:

$$\begin{cases} \text{If } z \_ z_{th} & ; \text{accept } H_0; \\ \text{Otherwise} & ; \text{accept } H_1; \end{cases} \quad \_ (2)$$

Where the threshold z_{th} is the solution to the equation $h_0(z_{th}) = h_1(z_{th})$. Under this strategy, P_{fa} and P_{md} are the areas of the shaded regions shown in Fig. 3a, respectively. The problem with this mechanism is that, when the mean of y is small, $h_1(z)$ and $h_0(z)$ are not sufficiently separated, leading to large P_{fa} and P_{md}, as shown in Fig. 3b.



a) Mean of v much greater than x b) Mean of v comparable to v

Fig. 3.Insufficiency of conventional detection algorithms when malicious packet drops are highly selective.

This observation implies that when malicious packet drops are highly selective, counting the number of lost packets is not sufficient to accurately differentiate between malicious drops and link errors. For such a case, we use the correlation between lost packets to form a more informative decision statistic. 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 s1; . . . ; sM for M independent messages r1; . . . ; rM, respectively. The source sends out the ri's and si'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 ci'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 s1; . . . ; sM.

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 signatures1; . . . ; sM.

Our construction ensures that si and ri are sent together along the route, so that knowledge of s1; . . . ; sM also proves that the node must have received r1; . . . ; rM. Our detection architecture consists of four phases: setup, packet transmission, audit, and detection.

IV. Intrusion Detection Scheme For Source Node Misbehavior

Numerous schemes have been proposed for secure routing and Intrusion Detection for ad hoc networks. Yet, little work exists in actually implementing such schemes on small

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

handheld devices. In this paper, we present a proof-of-concept implementation of a secure routing protocol based on AODV over IPv6, further reinforced by a routing protocol independent Intrusion Detection System (IDS) for ad hoc networks. Security features in the routing protocol include mechanisms for non-repudiation and authentication, without relying on the availability of a Certificate Authority (CA) or a Key Distribution Center (KDC). We present the design and implementation details of our system, the practical considerations involved, and how these mechanisms can be used to detect and thwart malicious attacks. We discuss several scenarios where the secure routing and intrusion detection mechanisms isolate and deny network resources to nodes deemed malicious.

V. Reducing Computation Overhead: 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 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.

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 nj generates a block reception bitmap bj= (bj1; . . . ; bjM), where bji=1 if and only if all L packets in block Bi has been received by nj, and bji= 0 otherwise. Except the above, Ad still follows the same algorithm 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.

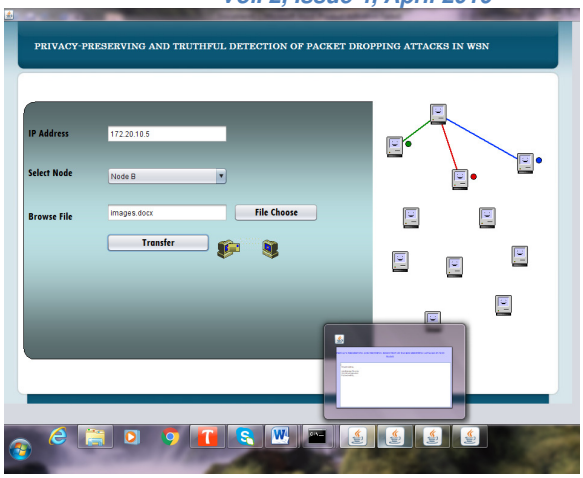
VI. Performance Evaluations

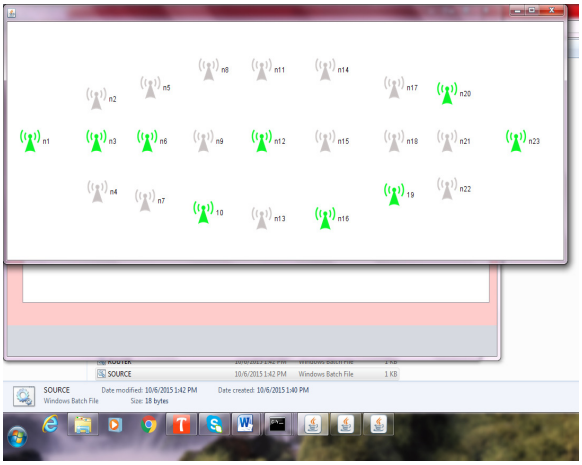
In this section, we compare the detection accuracy achieved by the proposed algorithm with the optimal maximum likelihood algorithm, which only utilizes the distribution of the number of lost packets. For given packet-loss bitmaps, the detection on different hops is conducted separately. So, we only need to simulate the detection of one hop to evaluate the performance of a given algorithm. We assume packets are transmitted continuously over this hop, i.e., a saturated traffic environment. We assume channel fluctuations for this hop follow the Gilbert-Elliot model, with the transition probabilities from good to bad and from bad to good given by PGB and PBG, respectively. We consider two types of malicious packet dropping: random dropping and selective dropping. In the random dropping attack, a packet is dropped at the malicious node with probability PM. In the selective dropping attack, the adversary drops packets of certain sequence numbers. In our simulations, this is done by dropping the middle N of the M most recently received packets, i.e., setting the N bits in the middle of the packet-loss bitmap to 0 (if a packet in these positions is dropped due to link errors, then the set of 0's extends to an extra bit in the middle). PM and N are simulation parameters that describe the selectivity of the attack. In both cases, we let $€^{th}$ = 10% for the proposed algorithm. We are interested in the following three performance metrics: probability of false alarm (Pfa), probability of miss detection (Pmd), and the overall detection-error probability (Perror). We collect these statistics as follows. In each run, we first simulate 1,000 independently generated packet-loss bitmaps for the hop, where packet losses are caused by link errors only. We execute our detection algorithm over these packet-loss bitmaps and collect the number of cases where the algorithm decides that an attacker is present. Let this number be Ifa. Pfa of this run is calculated as Pfa = Ifa= 1; 000. We then simulate another 1,000 independently generated packet-loss bitmaps, where losses are now caused by both link errors and malicious drops. Let the number of cases where the detection algorithm rules that an attacker is not present are Imd. Pmd of the underlying run is given by Pmd = Imd=1; 000. Perror is given by Perror= (Ifa + Imd) = 2; 000. The above simulation is repeated 30 times, and the mean and 95 percent confidence interval are computed for the various performance metrics.

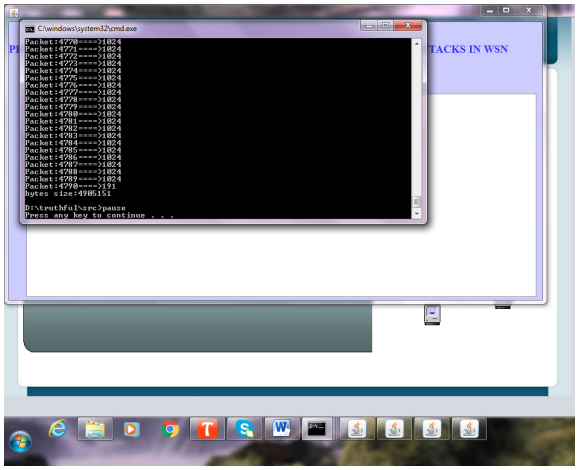
ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

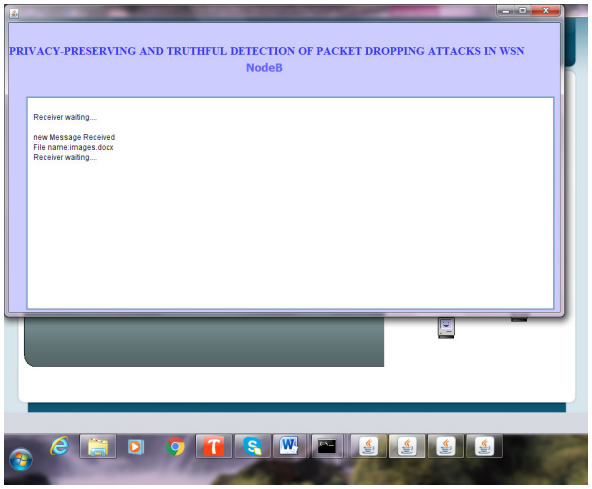
IJARMATE
Your idea MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016









VII. 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.

REFERENCES

[1] 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. Secure. Oct. 2007, pp. 598–610.

[2] 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

[3] 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. Inform.Syst. Security, vol. 10, no. 4, pp. 1–35, 2008.

[4] 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.

[5] K. Balakrishnan, J. Deng, and P. K. Varshney, "TWOACK: Preventing selfishness in mobile ad hoc networks," in Proc. IEEEWireless Commun. Netw. Conf., 2005, pp. 2137–2142.

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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

[7] 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.

[8] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring, "Modeling incentives for collaboration in mobile ad hoc networks," presented at the First Workshop Modeling Optimization Mobile, AdHoc Wireless Netw., Sophia Antipolis, France, 2003.

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

[10] 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.

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

[12] 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.

[13] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks, "in Proc. ACM MobiHoc Conf., 2005, pp. 46–57.

[14] Y. Zhang, L. Lazos, and W. Kozma, "AMD: Audit-based misbehavior detection in wireless ad hoc networks," IEEE Trans. MobileComput., PrePrint, Vol. 99, published online on 6 Sept. 2013.