JES
Journal of Engineering Sciences

Vol 10, Issue 12, Dec /2019
ISSN NO:0377-9254

Congestion Control Through IDA Process for Malicious Node in Disruption Attacks

SEYED AMIN AHMADI OLOUNABADI[1], AVULA DAMODARAM[2], V KAMAKSHI PRASAD[3], PVS SRINIVAS[4]

[1]Research scholar, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana.

[2]Professor, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana

[3]Professor & Director of DE, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana

[4]Professor, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana

Email: saminahmadi@hotmail.com

Abstract:

Sensor networks enable a wide range of applications in both military and civilian domains. However, the deployment scenarios, the functionality requirements, and the limited capabilities of these networks expose them to a wide-range of attacks against control traffic (such as wormholes, rushing, Sybil attacks, etc) and data traffic (such as selective forwarding). In this paper we propose a framework called UNMASK that mitigates such attacks by detecting, diagnosing, and isolating the malicious nodes. UNMASK uses as a fundamental building block the ability of a node to oversee its neighboring nodes' communication. On top of UNMASK, we build a secure routing protocol, LSR, that provides additional protection against malicious nodes by supporting multiple node-disjoint paths. We analyze the security guarantees of UNMASK and use ns-2 simulations to show its effectiveness against representative control and data attacks. The overhead analysis we present shows that UNMASK is a lightweight protocol appropriate for securing resource constrained sensor networks.

Keywords: sensor network security, neighbor monitoring, secure routing, control attack, data attack

1. Introduction:

Wireless sensor networks are emerging as a promising platform that enable a wide range of applications in both military and civilian domains such as battlefield surveillance, medical monitoring, biological detection, home security, smart spaces, inventory tracking, etc. Such networks consist of small, low-cost, resource-limited (battery, bandwidth, CPU, memory) nodes that communicate wirelessly and cooperate to forward data in a multi-hop fashion. Thus, they are especially attractive in scenarios where it is infeasible or expensive to deploy a significant networking infrastructure. However, the open nature of the wireless communication, the lack of infrastructure, the fast deployment practices, and the hostile deployment environments, make sensor networks vulnerable to a wide range of security attacks targeting the control or data traffic. Coping with control and data attacks in sensor networks is more challenging than in ad hoc wireless and

wired networks due to the resource constrained environment.

Typical examples of control traffic (numbers start with C for control) are routing, monitoring whether a node is awake, asleep, or dead, topology discovery, and distributed location determination. Control traffic attacks include the (Ci) wormhole attack ([16],[17]), (Cii) the rushing attack [18], (Ciii) the Sybil attack [11], (Civ) the sinkhole attack [14], and (Cv) the HELLO flood attack [14]. Control attacks are especially dangerous because they can be used to subvert the functionality of the routing protocol and create opportunities for a malicious node to launch data traffic attacks such as dropping all or a selective subset of data packets.

In addition to control traffic attacks, sensor networks are also vulnerable to data traffic attacks (numbers starts with D for data). The most notable data traffic attacks are (Di) blackhole, (Dii) selective forwarding and (Diii) artificially delaying of packets, in which respectively a malicious node drops data (entirely or selectively) passing through it, or delays its forwarding. The attacks could result in a significant loss of data or degradation of service.

The focus of this paper is on proposing mitigation techniques for control and data attacks in sensor networks. We present a lightweight framework called UNMASK (Utilizing Neighbor Monitoring for Attacks Mitigation in Multihop Wireless Sensor Networks), which mitigates control and data traffic attacks in sensor networks. UNMASK not only detects the occurrence of an attack, but also diagnoses the malicious nodes involved in it and removes their capability of launching future attacks by isolating them from the network. The detection and isolation mechanisms are executed locally, without incurring a significant overhead. UNMASK is suited to the low cost point of sensor networks since it does not require any specialized hardware (such as directional antennas [17] or GPS) nor does it require time-synchronization among the nodes [16]. UNMASK achieves its security goals by exploiting a well-known technique whereby nodes oversee part of the traffic going in and out of their neighbors [15], [25], [30], [31]. In our work, we present the technique in a formal framework–local monitoring–, identify the parameters that affect its performance, and analyze its capabilities and limitations. UNMASK can be applied to mitigate any control or data traffic attack that exploits one or more of the basic malicious primitives- drop, delay, fabricate, and modify. However, we exemplify the fundamental structures and the state to be maintained at each node for mitigating some representative attacks – Sybil, wormhole, rushing, and selective forwarding attacks. The first three are examples of attacks directed to control traffic while the last one is an example directed at data traffic. Independent of the detection mechanism, we propose a strategy to isolate malicious nodes locally in a distributed manner.

We use UNMASK to create a novel lightweight secure routing protocol called LSR that withstands known attacks against the routing infrastructure and provides additional protection against data attacks by supporting secure node-disjoint multiple route discovery. We analyze the detection coverage and the probability of false detection of UNMASK. We also evaluate the memory, communication, and computation overhead of UNMASK. Finally, we simulate the wormhole attack in ns-2 and show its effect on the network performance with and without UNMASK. The results show that UNMASK can achieve 100% detection of the wormholes for a wide range of network densities. They also show that the detection and isolation of the nodes involved in the wormhole can be

achieved in a fairly short time after an attack starts. In addition, we simulate a combined Sybil and rushing attack to bring out the adverse impact on node-disjoint multipath routing and show the improvement using UNMASK. The results show that LSR using UNMASK is resilient to the combined attack and that the average number of node-disjoint routes discovered is not reduced. Our experiments with data monitoring show the feasibility of detecting the selective forwarding attack while monitoring only a fraction of the data traffic. In summary, our contributions are as follows:

- We propose a mechanism to detect any control or data attack that results from dropping, delaying, modifying, or fabricating of packets.
- We develop a toolset based on overheard information that can be mapped to detecting different classes of attacks. We analyze this toolset for different metrics, such as, false alarm probability, missed alarm probability, and latency of isolation.
- We propose a mechanism that, based on information collected by our toolset, allows for diagnosing and isolating the malicious nodes.
- We demonstrate the effectiveness of our toolset applied to both data and control attacks through simulations.

This work builds on and extends our previous work in applying local monitoring as presented in [36] and [37].

2. System Model and Assumptions

Attacker model: An attacker can control an external node (i.e., a node that does not know the cryptographic keys that allows it to be authenticated by the rest of the nodes), or an internal node, (i.e., a node that possesses all the keys required for it to be authenticated by other nodes in the network, but exhibits malicious behavior). An insider node may be created, for example, by compromising a legitimate node. A malicious node can perform all the attacks mentioned in Section 1, by itself or by colluding with other nodes. However, we do not address the misrouting attack in which the attacker incorrectly forwards packets nor we consider the denial of service attacks. A malicious node can establish out-of-band fast channels (e.g., a wired link) or have a high powered transmission capability.

System assumptions: We assume that all the communication links are bi-directional. A finite amount of time is required from a node's deployment for it to be compromised, and to perform the first- and second-hop neighbor discovery protocol. We assume that no external or internal malicious nodes exist before the completion of the neighbor discovery. However, we can remove this assumption and use one of the protocols for secure neighbor discovery such as the directional antenna by Hu and Evans [17] at the additional cost of using directional antennas or by using trusted and more powerful nodes as in [42]. In our protocol we call the sensor node a guard when performing traffic overhearing and monitoring of neighbors. We assume that the network has sufficient redundancy, such that each node has more than an application defined threshold number of legitimate nodes as guards. We assume that the network has a static topology. This does not rule out route changes due to natural and malicious node failures or route evictions from the routing cache. Moreover, we assume that each node explicitly announces the immediate source of the packet it is forwarding. Finally, we assume a key management protocol, e.g., [22], is used to pre-distribute pair-wise keys such that any two nodes in the network can securely communicate with each other.

3 Primitives: Neighbor Discovery and One Hop Source Authentication

Neighbor discovery: This protocol is used to build a data structure of the first hop

neighbors of each node and the neighbors of each neighbor. The data structure is used in local monitoring to detect malicious nodes and in local response to isolate these nodes. A neighbor of a node, W, is any node that lies within the transmission range of W. As soon as a node, say A, is deployed in the field, it sends a one-hop broadcast of a HELLO message. Any node that receives the message sends a reply back to A. For each reply received within a pre-defined timeout (TROUT,), A adds the responder to its neighbor list, RA. Let RA = W1, ..., Wp and Msg = RA||Kcommit(A), where Kcommit(A) is the commitment key A uses later to authenticate itself to its neighbors. Node A then sends a one-hop broadcast of Msg. A node Wj that receives Msg, stores RA (Wj's second-hop neighbors) and Kcommit(A). Hence, at the end of this neighbor discovery process, each node has a list of its direct neighbors and their neighbors as well as the commitment key of each one of its direct neighbors. This process is performed only once in the lifetime of a node and prevents incorrect neighborhood membership in static wireless networks that follow our assumptions of attack-free environment during neighbor discovery.

Commitment key generation and update: This protocol is used to generate and update the commitment key used by the one-hop source authentication protocol. The values of the commitment key at a node S (Kcommit(S)) are derived from a random seed (Kseed(S)) as Kcommit(S) = H(i) (Kseed(S)), where H is a one-way collision resistant hash function [45][47], i takes values between 0 and l(≥2), and l is the length of the sequence of values of Kcommit(S) that we call the commitment string. The first value of the commitment key Kcommit(S) that is exchanged with the neighbors during neighbor discovery is H(l)(Kseed(S)) = vl. The subsequent values of the commitment key (vl-1,…, v0) are progressively disclosed to the neighbors during subsequent transmissions. Before the current commitment string {vl, vl-1,…, v0} is exhausted, a new one is generated at S {ul, ul-1,…,u0}. The commitment key ul from the new string is authenticated to the neighbors using the last undisclosed key from the current string with the one-hop source authentication protocol.

One-hop source authentication: This protocol allows a node to distinguish between its neighbors to prevent identity spoofing among them. A node S authenticates its transmitted packets to the neighbors by attaching the last undisclosed value from the commitment string Kcommit(S). This authentication is only used with the source of the packet, not at every hop in the path of the packet from the source to the destination . When a neighbor of S, say B, receives the packet, it verifies the validity of Kcommit(S) by computing a hash function over it and comparing the result with the stored value of Kcommit(S). If Kcommit(S) is valid, B stores it as the new commitment key value of S. However, this protocol may fail to provide the required authentication if an attacker blocks the transmission range of a certain source from the rest of network except itself. Therefore, the attacker can impersonate that source and generate valid packets. In such case, we revert to the well-known μTESLA authentication scheme [21] which countermeasures such attacks.

4. Results and Discussions:

This paper uses the ns-2 simulator [29] to simulate a data exchange protocol over LSR, individually without UNMASK (the baseline) and with UNMASK. We distribute the nodes randomly over a square area with a fixed average node density. Thus, the length of the square varies (80m to 204m) with the number of nodes (20-250).

This random distribution may result in situations where the number of good guards of some nodes goes below γ, which negatively impact the simulation results. We first simulate the wormhole attack using out-of-band direct channels between the colluding nodes. The malicious nodes are randomly selected from the network nodes. After a wormhole is established, the malicious nodes at each end of the wormhole drop all the packets forwarded to them.

Each node acts as a source and generates data according to a Poisson process with rate μ. The destination is chosen at random and is changed using an exponential random distribution with rate ξ. A route is evicted if unused for TOutRoute time. Isolation latency is defined as the time between when the node performs its first malicious action to the time by which all the neighbors of the node have isolated it. The experimental parameters are given in Table 1. The results are averages over 30 runs. The malicious nodes are chosen at random such that they are more than 2 hops away from each other.

Table 1: Input parameter values

Parameter	Value	Parameter	Value
Tx Range (r)	30 m	γ	2-8
N_B	8	μ	100
$TOut_{Route}$	50 sec	*M*	0-10
τ , N_r	0.05 s, 5	β	5
Channel BW	40 kbps	ξ	5

Figure 1 shows the number of packets dropped as a function of simulation time for the 100-node setup with 2 and 4 colluding nodes. The attack is started 50 seconds after the start of the simulation. Since the numbers are vastly different in the baseline and with UNMASK, they are shown on separate Y-axes (the left corresponding to the Baseline and the right corresponding to the UNMASK case). In the baseline case, since wormholes are not detected and isolated, the cumulative number of packets dropped continues to increase steadily with time. But in UNMASK, as wormholes are identified and isolated permanently, the cumulative number stabilizes. Note that the cumulative number of packets dropped grows for some time even after the wormhole is locally isolated. This is because the cached routes that contain the wormhole continue to be used until route timeout occurs.

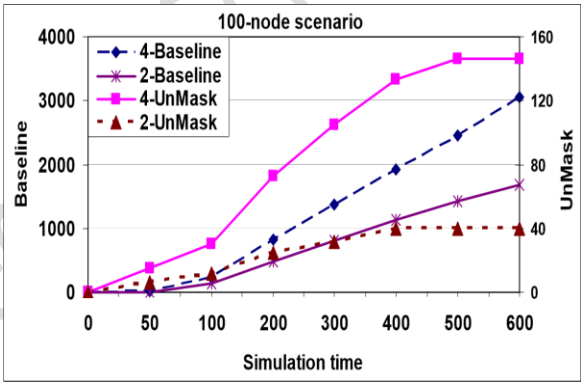

Figure 1: Cumulative no. of dropped packets

Conclusion:

We have presented a distributed protocol, called UNMASK, for detection, diagnosis, and isolation of nodes launching control attacks, such as, wormhole, Sybil, rushing, sinkhole, and replay attacks. UNMASK uses local monitoring to detect control and data traffic misbehavior, and local response to diagnose and isolate the suspect nodes. We analyze the security guarantees of UNMASK and show its ability to handle attacks through a representative set of these attacks. We present a coverage analysis and find the probability of false alarm and missed detection. On top of UNMASK, we build a secure lightweight routing protocol, called LSR, which also supports node disjoint path discovery. We note that although designed for static networks, UNMASK can potentially be

extended to mobile networks. In mobile networks the neighborhood changes and therefore the neighbor discovery is required to be executed during the lifetime of the network. Therefore, the neighbor discovery protocol presented here cannot be secure for mobile networks. Note that incremental deployment of nodes is equivalent to a node moving to the new position and the situation can be handled similarly. As future work we are investigating secure neighbor discovery protocols appropriate for resource-constrained mobile networks.

References:

[1] M. G. Zapata, Secure ad-hoc on-demand distance vector (SAODV) routing, IETF MANET Mailing List, October 8, 2001.

[2] Y.-C. Hu, D. B. Johnson, and A. Perrig, SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks, WMCSA 2002, pp. 3-13.

[3] Y.-C. Hu, A. Perrig, and D. B. Johnson, Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks, MobiCOM 2002, pp. 12-23.

[4] P. Papadimitratos and Z. Haas, Secure routing for mobile ad hoc networks, SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), January 2002.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, MobiCOM 2000, pp. 56-67.

[6] B. Karp and H. T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, MobiCOM 2000, pp. 243-254.

[7] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, Highly-resilient, energy-efficient multipath routing in wireless sensor networks, Mobile Computing and Communications Review, vol. 4, no. 5, October 2001, pp. 1125.

[8] F. Ye, A. Chen, S. Lu, and L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, ICCCN 2001, pp. 304-309.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy efficient communication protocol for wireless micro sensor networks, HICSS 2000, pp. 3005-3014.

[10] D. Braginsky and D. Estrin, Rumor routing algorithm for sensor networks, WSNA 2002, pp. 22-31.

[11] J. Newsome, E. Shi, D. Song, and A. Perrig, The Sybil attack in Sensor Networks: Analysis & Defenses, IPSN 2004, pp. 259-268.

[12] K. Ishida, Y. Kakuda, and T. Kikuno, A routing protocol for finding two node-disjoint paths in computer networks, ICNP 1992, pp. 340 347.

[13] Y. Xu, J. Heidemann, and D. Estrin, Geography-informed energy conservation for ad hoc routing, MobiCOM 2001, pp. 70-84.

[14] C. Karlof and D. Wagner, Secure Routing in Sensor Networks: Attacks and Countermeasures, SNPA 2003, pp. 113-127.

[15] S. Marti, T. J. Giuli, K. Lai, and M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, MobiCOM 2000, pp. 255-265.

[16] Y. C. Hu, A. Perrig, and D.B. Johnson, Packet leashes: a defense against wormhole attacks in wireless networks, IEEE INFOCOM 2003, pp. 1976- 1986.

[17] L. Hu and D. Evans, Using Directional Antennas to Prevent Wormhole attacks, NDSS 2004, pp. 131-141.

[18] Y. C. Hu, A. Perrig, and D. Johnson, Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols, WiSe 2003, pp.30-40.

[19] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer, A Secure

Routing Protocol for Ad hoc Networks, ICNP 2002, pp. 78-87.
[20] S. Lindsey and C. Raghavendra, PEGASIS: power-efficient gathering in sensor information systems, IEEE Aerospace Conference 2002, vol. 3, 1125 - 1130.
[21] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, SPINS: Security Protocols for Sensor Networks, Wireless Networks 2002., vol. 8, pp. 521-534
[22] D. Liu and P. Ning, Establishing Pair-wise Keys in Distributed Sensor Networks, CCS 2003, pp. 52-61.
[23] C. E. Perkins and E. M. Royer, Ad-Hoc On-Demand Distance Vector Routing, in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99) 1999, pp. 90-100.
[24] P. Papadimitratos and Z.J. Haas, Secure Message Transmission in Mobile Ad Hoc Networks, WiSe 2003, pp.41-50.
[25] S.J. Lee and M. Gerla, Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks, ICC 2001, pp. 3201-3205.
[26] A. Nasipuri, R. Castaneda, and S.R. Das, Performance of Multipath Routing for On-demand protocols in Mobile Ad Hoc Networks, ACM Mobile Networks and Applications (MONET) 2001, 6(4): pp. 339-349.
[27] Z. Ye, S. V. Krishnamurthy, S. K. Tripathi, A Framework for Reliable Routing in Mobile Ad Hoc Networks, IEEE INFOCOM 2003, vol.1, pp. 270-280.
[28] G. Bianchi, Performance analysis of the IEEE 802.11 Distributed Coordination Function, IEEE Journal on Selected Areas in Communications, 2000, 18(3): pp. 535-547.
[29] The Network Simulator ns-2, At: www.isi.edu/nsnam/ns/
[30] A. A. Pirzada and C. McDonald, Establishing Trust In Pure Ad-hoc Networks, Proceedings of 27th Australasian Computer Science Conference (ACSC'04), pp. 47-54.
[31] S. Buchegger, J.-Y. Le Boudec, Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Distributed Ad-hoc NeTworks, in MobiHoc 2002, pp. 80-91.
[32] Y. Huang and W. Lee, A Cooperative Intrusion Detection System for Ad Hoc Networks, SASN 2003, pp. 135147.
[33] B. Schneier, Applied Cryptography, 2nd edition, Prentice Hall, 1996.
[34] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, Y. C. Hu, TIBFIT: Trust Index Based Fault Tolerance for Arbitrary Data Faults in Sensor Networks, DSN 2005, pp. 672 - 681.
[35] Secure Hash Standard, Federal Information Processing Standards Publication 180-1, April 1995.
[36] I. Khalil, S. Bagchi, and N. B. Shroff, LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks, DSN 2005, pp. 612-621.
[37] I. Khalil, S. Bagchi, and C. Nita-Rotaru, DICAS: Detection, Diagnosis and Isolation of Control Attacks in Sensor Networks, SecureComm 2005.