AN EFFECTIVE INTRUDER DETECTION WITH BACKUP TIME SERIES TECHNIQUES IN WIRELESS NETWORK

Dr.D.Devakumari[1], K.Gayathri[2]

[1] Assistant Professor, PG and Research Department of Computer Science, Government Arts College(Autonomous), Coimbatore, Tamil Nadu, India
[2] Research Scholar ,Department of Computer Science , L.R.G Government Arts College For Women , Tirupur, Tamil Nadu, India

---

Abstract - In wireless network, every node accesses the network in a cooperative manner and randomly delays transmissions to avoid collisions by following a common backoff rule [1]. However, in such a distributed environment without a centralized controller, a malicious node may deliberately choose a smaller backoff timer and selfishly gain an unfair share of the network throughput at the expenses of other normal nodes' channel access opportunities. The distributed nature of the CSMA/CA-based wireless protocols allows malicious nodes to deliberately manipulate their backoff parameters and, thus, unfairly gain a large share of the network throughput. While most of the existing schemes for selfish misbehavior detection depend on heuristic parameter configuration and experimental performance evaluation, the projects develop a Markov chain-based analytical model to systematically study the performance of the FS detector in real-time backoff misbehavior detection.

Key Words: Denial of service, security, routing, ad hoc networks, sensor networks, wireless networks

1. Introduction

AD hoc wireless sensor networks (WSNs) promise existing new applications in the near future, such as ubiquitous on-demand computing power, continuous connectivity, and instantly deployable communication for military and first responders. Such networks already monitor environmental conditions, factory performance, and troop deployment, to name a few applications. As WSNs become more and more crucial to the everyday functioning of people and organizations, availability faults become less tolerable—lack of availability can make the difference between business as usual and lost productivity, power outages, environmental disasters, and even lost lives; thus high availability of these networks is a critical property, and should hold even under malicious conditions. Due to their ad hoc organization, wireless ad hoc networks are particularly vulnerable to denial of service (DoS) attacks , and a great deal of research has been done to enhance survivability [2], [5], [13], [14]. While these schemes can prevent attacks on the short term availability of a network, they do not address attacks that affect long-term availability—the most permanent denial of service attack is to entirely deplete nodes' batteries. This is an instance of a resource depletion attack, with battery power as the resource of interest. In this paper, we consider how routing protocols, even those designed to be secure, lack protection from these attacks, which we call Vampire attacks, since they drain the life from networks nodes. These attacks are distinct from previously studied DoS, reduction of quality (RoQ), and routing infrastructure attacks as they do not disrupt immediate availability, but rather work over time to entirely disable a network. While some of the individual attacks are simple, and power draining and resource exhaustion attacks have been discussed before prior work has been mostly confined to other levels of the protocol stack, e.g., medium access control (MAC) or application layers, and to our knowledge there is little discussion, and no thorough analysis or mitigation, of routing-layer resource exhaustion attacks. Vampire attacks are not protocol-specific, in that they do not rely on design properties or implementation faults of particular routing protocols, but rather exploit general properties of protocol classes such as link-state, distancevector, source routing, and geographic and

beacon routing. Neither do these attacks rely on flooding the network with large amounts of data, but rather try to transmit as little data as possible to achieve the largest energy drain, preventing a rate limiting solution. Since Vampires use protocol-compliant messages, these attacks are very difficult to detect and prevent.

2.Related Work

The problem of detecting backoff misbehavior over the 802.11-based medium access control (MAC) protocol has been widely studied in the literature. In [8], [9], a modification to the 802.11 protocol is proposed to facilitate the misbehavior detection, where the receiver assigns a backoff timer for the sender. If the number of idle slots between consecutive transmissions from the sender does not comply with the assigned backoff timer, the receiver may label the sender as a selfish node. Modification to the 802.11 protocol and reliance on a trustworthy receiver are the main limitations of the work. Another approach to deal with the backoff misbehavior is to develop protocols based on the game-theoretic techniques [14], [15], [16]. The goal is to encourage all the nodes to reach a Nash equilibrium. As a result, a malicious node is not able to gain an unfair share compared to wellbehaved nodes and, thus, discouraged from the misbehavior. However, this category of approaches assumes that all the nodes are willing to deviate from the protocol when necessary, and the standard protocol needs to be modified. A heuristic sequence of conditions is proposed in [17], [18] to test multiple misbehavior options over the 802.11 MAC based on simple numerical comparisons. This approach, named DOMINO, preserves its advantage of simplicity and easiness of implementation, and still demonstrates its efficiency when dealing with a wide range of 802.11 MAC misbehavior. However, the heuristic nature of the approach limits its applications to specific scenarios. The sequential probability ratio test (SPRT) method is used in [5], [6], [7] to detect the 802.11 backoff misbehavior. The detection decision is made when a random walk of the likelihood ratio of observations (given two hypotheses) rises to be larger than an upper threshold. The main advantage of SPRT is that it can reach decision very fast, given the complete knowledge of both normal behavior and backoff misbehavior strategy [20]. However, in a realistic setting, the strategy of malicious nodes is hard to know in advance. Further, the existing work normally assumes that the backoff timer of each node is observable, which is again hard to achieve in practice because thetransmission attempts involved in a collision are impossible to be distinguished. In our design, we monitor the successful transmission of the tagged node as the observation measurement.

The authors in [3], [4] utilize the Kolmogorov-Smirnov significance test for backoff misbehavior detection. This test is able to make the decision by measuring the distribution of the idle time between consecutive successful transmissions from a tagged node and comparing it to the normal backoff behavior. The detection method in [3], [4] requires estimation of the collision probability of a packet transmitted. However, an inaccurate simplification there is to consider that packets from the misbehaving node and those from the normal nodes have the same collision probability. Such inaccuracy impacts both the performance of false positive rate and detection delay, to be demonstrated in Section 7. Furthermore, as a batch test method, the K-S statistic has its own drawback. Fixed-size data samples are needed to perform the test each time, which makes real-time detection difficult. In our preliminary work [19], we adopt the nonparametric CUSUM test [12] for the backoff misbehavior detection, which has the advantages of both real-time detection and no requirement of a priori knowledge of the misbehavior strategy. The detector in [19] directly counts the number of successful transmissions from a tagged node within an observation window1 to get a sample. Although such a sampling method is easy for implementation, the observation window needs to linearly increase with the number of nodes in the network to fairly count transmissions from each node, which as a result will increase the detection delay. In this paper, we develop the new FS detector, which takes every successful transmission over the network as a sample to trigger its state change. Such a sampling method is independent of the network size and turns out to result in good performance in both false positive rate and detection delay, as to be demonstrated later in this paper.

A common research issue among most of the existing schemes for misbehavior detection is their dependency on heuristic parameter configuration and experimental performance evaluation, which largely limits the flexibility and robustness of the schemes. To address this issue, in [19], we propose a Markov chain-based analytical model to theoretically analyze the detection performance and quantitatively configure the system parameters. In this paper, we develop the analytical model according to the newly proposed FS detector. Our analysis demonstrates performance improvement of the FS detector in real-time misbehavior detection over the original detector in [19]. Also, we demonstrate the robustness of the FS detector under varying network size, against the short-term unfairness, and in the situation when both UDP and TCP traffic exists.

3.Methods

3.1 Misbehave Action Occurrence and Finding the Action

In this module, the node sends the data repeatedly even the receiver is not able to receive quickly. To find the action, a real-time back-off misbehavior detector, termed as the fair share detector (FS detector), which exploits the nonparametric cumulative sum (CUSUM) test to quickly find a selfish malicious node without any a priori knowledge of the statistics of the selfish misbehavior is applied. The **observation measure** is based on the following.

Consider a tagged node v. In the detection system, the observation measure is an indicator of whether a successful transmission over the network belongs to the tagged node v, denoted as Iv

We take the popular modeling technique that each node independently accesses an idle channel for transmission with a probability determined by its contention window size. If we use qv s to denote the probability that a successful transmission over the network is from node v, the probability distribution of Iv

$$P\{I^v = k\} = \begin{cases} q_s^v & \text{if } k = 1, \\ 1 - q_s^v & \text{if } k = 0. \end{cases}$$

is given by

In a normal situation that every node follows the 802.11 DCF standard, it can be seen that $q^v_s = 1/N$ due to fair channel sharing, given N nodes in the network. If node v is a malicious node taking a smaller contention window size, it will achieve a q^v_s larger than 1/N and, thus, a larger portion of the network throughput.

The **Fair Share Detector** is based on the following.

Let {In, n = 0, 1, ...} be the sequence of sample values of Iv, observed each time a successful transmission appears on the channel. Here, we drop the superscript v for easier presentation considering the clear context. There are N nodes and one access point (AP) in the network. Suppose that the initial value of the detector is $X_0 = 0$. If a successful transmission upon the nth observation is from the tagged node, i.e., $I_n = 1$, the detector X_n increases by N – 1, otherwise, $I_n = 0$, and X_n decreases by 1 until it reaches 0.

The intuition of this design is as follows: In the normal situation, each node roughly takes turn to transmit; the increase of X_n caused by one successful transmission from the tagged node can then be equally offset by the successful transmissions from other N - 1 nontagged nodes. Thus, the detector Xn will fluctuate around a low value close to zero in the normal situation.

On the other hand, when the tagged node turns to misbehave and obtain more chances to transmit, it is not difficult to see that X_n is going to quickly accumulate to a large positive value. The behavior of the FS detector can be mathematically described as

$$X_{n+1} = (X_n + (NI_n - 1))^+$$
$$X_0 = 0,$$

where $(x)^+ = x$ if x >= 0 or 0 otherwise. We can see that the above equation is actually in the form of a nonparametric CUSUM detector. Let h be the detection threshold. The decision rule of the detector in step n is

$$\delta_n = \begin{cases} 1 & \text{if } X_n \geq h, \\ 0 & \text{if } X_n < h, \end{cases}$$

where δ_n is also an indicator function of whether the detection event happens or not. The detector value X_n

will be reset back to 0 as soon as it exceeds the threshold and the detection procedure starts over again.

To implement the above concept, the receiver node checks for the sender's request continuously. If request is repeated with in the back off timer value, then it is found that the sender node is misbehaving and disturbing the receiver node. A threshold is set so that the request count (each request is sent before the backoff timer), is found to be above the threshold value, then the node is found to be misbehaving and alerted in receiver node.

3.2. Fixed Window Protocol Module

In monitoring-based intrusion detection, each node monitors the forwarding behavior of its neighboring nodes in fixed window mechanism. The node listens the next-hop node with the packets sent at regular intervals. If say ten packets (window frame size) are sent and listen for monitoring, then the packets dropped at the starting side of the window frame and the packets dropped at the ending side of the window frame are calculated. In addition, the n^{th} frame's ending side added with $n+1^{th}$ frame's starting side is added up.

If the value is above the given threshold value, then suspect count value is added with one. If the total suspect count is crossed the given suspect threshold value, then the node is suspected that it is dropping the packets suspiciously. Unlike the existing system, the node not only monitors the source node's windows frames but also monitors of the neighbor node which passes the packets to the suspected node.

3.3. Sliding Window Protocol Module

In monitoring-based intrusion detection, each node monitors the forwarding behavior of its neighboring nodes in sliding window mechanism. The node listens the next-hop node with the packets sent at regular intervals. If say ten packets (window frame size) are sent and listen for monitoring, the windows is calculated as 1^{st} frame to nth frame, then 2^{nd} frame to nth packet in first window and 1^{st} packet in second window and so on.

Then the packets dropped at the starting side of the window frame and the packets dropped at the ending side of the window frame are calculated. In addition, the n^{th} frame's ending side added with $n+1^{th}$ frame's starting side is added up.

If the value is above the given threshold value, then suspect count value is added with one. If the total suspect count is crossed the given suspect threshold value, then the node is suspected that it is dropping the packets suspiciously. Unlike the existing system, the node not only monitors the source node's windows frames but also monitors of the neighbor node which passes the packets to the suspected node.

4. EXPERIMENTAL RESULT

4.1 Comparison of Packet Drop between hope and multi hope protocols

The Packet drop count of existing hope base fixed and sliding window protocol is compared with the proposed Marko chain model for Multi-Hop Wireless Networks. The Packet drop count of existing protocol is drop count threshold 88 (ex: single hope). The packet drop count for the proposed protocol is 31(ex: multi hope). The comparison of packet drop count in FWP-SFP and MFWP-MSWP

PAC KETS	FWP-SWP OF PACKET DROP COUNT	MFWP-MSWP OF PACKET DROP COUNT
25	13	7
50	23	19
75	33	29
100	43	35

Table 4.1 Comparison of Packet Drop count of FWP-

SFP and MFWP-MSWP

The Packet drop count of existing hope base fixed and sliding window protocol is compared with the proposed Marko chain model for Multi-Hop Wireless Networks.

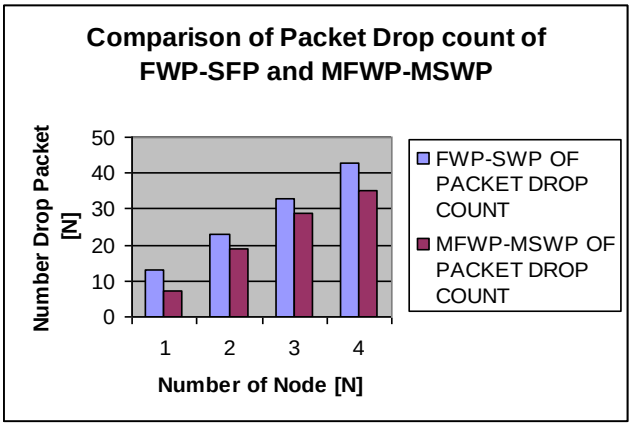

Fig 5.1 Comparison between Packet Drop counts

5.2.2. Performance analysis for existing system Cluster base Revocation Certification

The **Table 4.2** represents experimental result for existing system. The finding malicious node and revocation node process within second details and Mines details as followed.

S.NO	REVOCATION TIME (SEC)	NO.OF ATTACKER NODES	AVERAGE OF ATTACKER PER MINS (Throughput) (%)
1	100	125	3.68
2	200	195	10.67
3	300	356	25.38
4	400	384	38.22
5	500	475	60.41
6	600	566	90.63

Table 4.2 Experimental Result for number of node and average of attacker node finding in Existing System

The **Figure 4.2** represents experimental result for existing system. The finding malicious node and revocation node process within Minis details as followed.

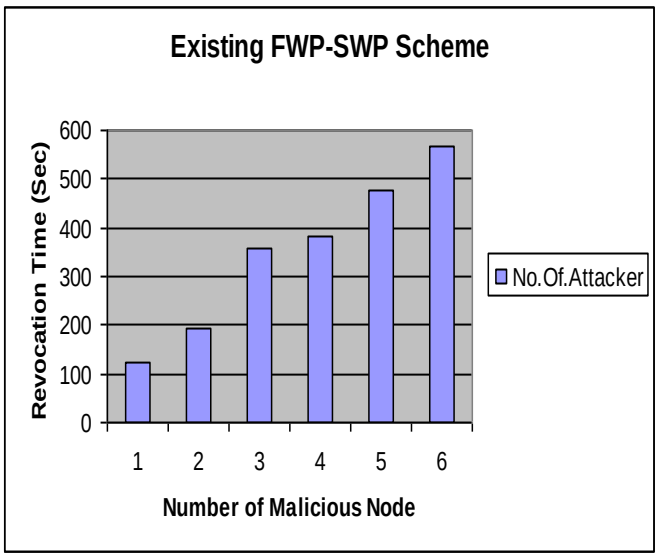

Fig 4.2 Existing FWP-SWP- Number of Attacker

The **Figure 4.3** represents experimental result for proposed system. The finding malicious node and revocation node process within Minis details as followed.

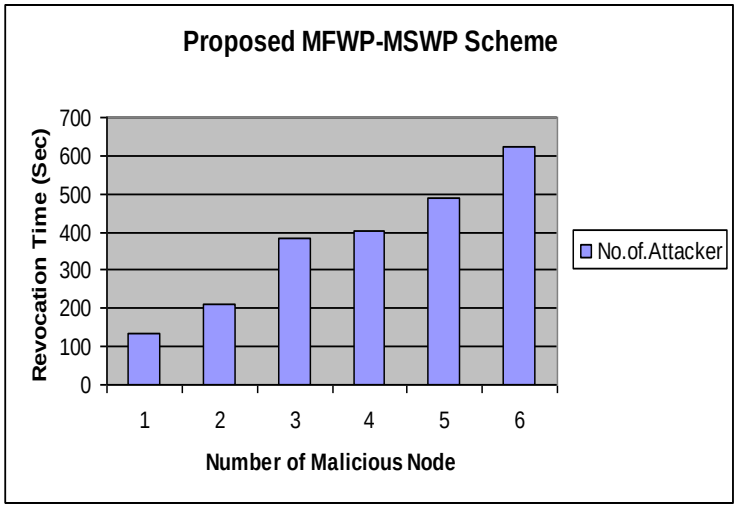

Fig 4.3 Proposed MFWP-MSWP - Number of Attacker

5. CONCLUSION

The new system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. In this project, major issues to ensure secure communications for mobile ad hoc networks, namely, certificate revocation of attacker nodes are solved. In contrast to existing algorithms, we propose a cluster-based certificate revocation with vindication capability scheme combined with the merits of both voting-based and non-voting based mechanisms to revoke malicious certificate and solve the problem of false accusation.

A new incentive method to release and restore the legitimate nodes and to improve the number of available normal nodes in the network has been proposed. This software is very particular in finding malicious applications. Any node with .Net framework installed can execute the application.

6. FUTURE ENHANCEMENT

The process of preparing plans had been a new experience, which was found use full in later phases of the project is completed. Efforts had been taken to make the system user friendly and as simple as possible. However at some points some features may have been missed out which might be considered for further modification of the application. The new system become useful if the below enhancements are made in future.

- Any attack should be identified as soon as possible.
- To mitigate malicious attacks on the network.

In future, the system is more effective and efficient in revoking certificates of malicious attacker nodes, reducing revocation time, and improving the accuracy and reliability of certificate revocation.

REFERENCES

[1] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in Mobile Ad Hoc Networks: Challenges and Solutions," IEEE Wireless Comm., vol. 11, no. 1, pp. 38-47, Feb. 2004.

[2] P. Sakari ndr and N. Ansari , "Security Services in Group Communications Over Wireless Infrastructure, Mobile Ad Hoc, and Wireless Sensor Networks," IEEE Wireless Comm., vol. 14, no. 5, pp. 8-20, Oct. 2007.

[3]A.M. Hegland, E. Winjum, C. Rong, and P. Spilling, "A Survey of Key Management in Ad Hoc Networks," IEEE Comm. Surveys and Tutorials, vol. 8, no. 3, pp. 48-66, Third Quarter 2006.

[4]L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," IEEE Network Magazine, vol. 13, no. 6, pp. 24-30, Nov./Dec. 1999.

[5] L. Zhou, B. Cchneider, and R. Van Renesse, "COCA: A Secure Distributed Online Certification Authority," ACM Trans. Computer Systems, vol. 20, no. 4, pp. 329-368, Nov. 2002.

[6] C. Gentry, "Certificate-Based Encryption and the Certificate Revocation Problem," EUROC RYPT: Proc. 22nd Int 'l Conf. Theory and Applications of Cryptographic Techniques, pp. 272-293, 2003.

[7] H. Yang, J. Shu, X. Meng, and S. Lu, "SCAN: Self-Organized Network-Layer Security in Mobile Ad Hoc Networks," IEEE J. Selected Areas in Comm., vol. 24, no. 2, pp. 261-273, Feb. 2006.

[8]J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Network: Analysis & Defenses," Proc. Third Int'l Symp. Information Processing in Sensor Networks, pp. 259-268, 2004.

[9] W. Liu, H. Nishiyama, N. Ansari, and N. Kato, "A Study on Certificate Revocation in Mobile Ad Hoc Network," Proc. IEEE Int'l Conf. Comm. (ICC), June 2011.

[10] P. Yi, Z. Dai, Y. Zhong, and S. Zhang, "Resisting Flooding Attacks in Ad Hoc Networks," Proc. Int'l Conf. Information Technology: Coding and Computing, 2005.

[11] Kong, X. Hong, Y. Yi, J.-S. Park, J. Liu, and M. Gerla, "A Secure Ad-Hoc Routing Approach Using Localized Self-Healing Com-munities," Proc. Sixth ACM Int'l Symp. Mobile Ad hoc Networking and Computing , pp. 254-265. 2005.

[12] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Net work Researc h," Wirele ss Comm. and Mobil e Computing (WCMC) Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications, vol. 2, no. 5, pp. 483-502, 2002.

[13] M. Sanchez and P. Manzoni. Anejos: A java based simulator for ad-hoc networks. Future Generation Computer Systems, 17(5):573–583, 2001.
[14] V. Davies. Evaluating mobility models within an ad hoc network. Master's thesis, Colorado School of Mines, 2000.
[15]E. W. Weisstein. The CRC Concise Encyclopedia of Mathematics. CRC Press, 1998.
[16] M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility pattern. IEEE Journal on Selected Areas in Communications, 15(7):1239–1252, 1997.
[17] I. Rubin and C. Choi. Impact of the location area structure on the performance of signaling channels in wireless cellular networks. IEEE Communications Magazine, pages 108–115, 1997.
[18] IEEE-SA Standards Board, "IEEE Std. 802.11i," IEEE, 2004. [2] ZigBee Alliance, "ZigBee Standard, version 1," ZigBee Alliance,2004.
[19] IEEE-SA Standards Board, "IEEE Std. 802.15.4," IEEE, 2003.
[20] Bluetooth SIG, "Bluetooth Core Specification Version 2.0 + Enhanced Data Rate," Bluetooth SIG, 2004.
[21]A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," Proc. CRYPTO '84, 1984, pp. 47–53. and Computing, pp. 254-265. 2005,
[22]Scalable Network Technologies: Qualnet, http://www.scalable-networks.com, 2012.
[23]C. Bettstetter, G. Resta, and P. Santi, "The Node Distribution of the Random Waypoint M obility Model for Wireless Ad Hoc Net-works," IEEE Trans. Mobile Computing, vol. 2, no. 3, pp. 257-269, July-Sept. 2003.
[24]J. Clulow and T. Moore, "Suicide for the Common Good: A New Strategy for Credential Revocation in Self-organizing Systems," ACMSIGOPS Operating Systems Rev., vol. 40, no. 3, pp. 18-21, July 2006.
[25]K. Park, H. Nishiyama, N. Ansari, and N. Kato, "Certificate Revocation to Cope with False Accusations in Mobile Ad Hoc Networks," Proc. IEEE 71st Vehicular Technology Conf. (VTC '10), May 16-19, 2010.
[26]E. Ayanoglu, C .- L. I, R. D. Gitlin, and J. E. Mazo. Diversity c o ding for transparent se lf-healing and fault-tolerant communication networks. IEEE Transactions on Communications , 41(11):1677–1686, Nove mb e r 1993.
[27]M. Castro and B. Liskov. Practical Byzantine fault tolerance. In Proceedings of the 3rd USE NIX Symposium on Operating System Design and Implementation (OSDI'99) , pages 173–186, New Orleans, LA USA, February 22–25, 1999. USENIX Association, IEEE TCOS, and ACM SIGOPS.
[28] C. Bettstetter, H. Hartenstein, and X. Perez-Costa. Stochastic Properties of the Random Waypoint Mobility Model. ACM/Kluwer Wireless Networks, Special Issue on Modeling and Analysis of Mobile Networks, 10(5):555–567, 2004.

Dr. D. Devakumari has received M. Phil degree from Manonmaniam Sundaranar University in 2003 and Ph.D from Mother Teresa Womens' University in 2013. Currently she is working as Assistant Professor in the PG and Research Department of Computer Science, Government Arts College (Autonomous), Coimbatore, India. Her research papers have been published in International journals including Inderscience, Springer etc. She has presented papers in National and International Conferences. Her research interests include Data Pre-processing and Pattern Recognition.

Mrs. K.Gayathri has received BCA degree from PKR Arts and Science College For Women and MCA from KSR College of Engineering .Pursuing her M.Phil degree from L.R.G Government College for Women.Currently she is working as Assistant Professor in Shiri Kumaran Arts & Science College Department of Computer Science, Coimbatore, India.