Defending against Collaborative Packet Drop Attacks on MANETs

Weichao Wang
Department of SIS
UNC Charlotte
Charlotte, NC 28223
wwang22@uncc.edu

Bharat Bhargava
CS Department
Purdue University
W. Lafayette, IN 47906
bb@cs.purdue.edu

Mark Linderman
Rome Lab
Air Force Research Lab
Rome, NY 13441
mark.linderman@rl.af.mil

***Abstract*—Detecting packet drop attacks is important for security of MANETs and current random audit based mechanism cannot detect collaborative attacks. In this paper, we design a hash function based method to generate node behavioral proofs that contain information from both data traffic and forwarding paths. The new method is robust against collaborative attacks described in the paper and it introduces limited computational overhead on the intermediate nodes. We investigate the security of the proposed approach and design schemes to further reduce the overhead.**

Keywords: collaborative packet drop attacks; hash based node behavioral proof; audit based detection;

I. INTRODUCTION

With the fast development and deployment of mobile devices, Mobile Ad Hoc Networks (MANETs) become an important component of modern distributed systems. Because of the infrastructure-less property, MANETs can be easily deployed. They are very attractive to applications such as military operations and first response to disasters. These applications, however, have very strict requirements on security of network topology and data traffic. Mechanisms must be properly designed for these applications before the advantages of MANETs can be fully exploited.

The security of MANETs has attracted a lot of research efforts and very encouraging results have been obtained. Most of the research efforts, however, focus on the prevention and detection of misbehaviors from individual attackers. Therefore, the effectiveness of these approaches will be weakened when adversaries work together to conduct collaborative attacks. For example, the WatchDog scheme proposed in [1] requires wireless nodes to monitor their neighbors to detect packet drop attacks. If multiple malicious nodes provide "evidences" to support each other's innocence, it will be very difficult to detect the sources of the black hole and grey hole. As another example, Packet Leash [2] uses accurate timestamps in packets to estimate the transmission distance and defend against wormhole attacks. If multiple attackers share their secret keys, the timestamp can be embedded and signed by the final sender in the wormhole and the tunneling behavior will not be detected. These examples show that collaborative attacks pose new challenges to security researchers.

In this paper, we propose to investigate the detection of collaborative packet drop attacks on MANETs. Several reasons lead us to the selection of this problem. First, since more and more applications in MANETs are becoming data-oriented, providing secure and robust data delivery becomes a top priority in protocol design. Second, random audits and node behavior monitoring can be used as a reactive approach to detecting packet drop attacks. In this way, we can reduce the overhead of the approach since it will be triggered only when the destination detects some anomaly in packet delivery ratio. Last but not least, the proposed approach is orthogonal to secure routing in MANETs and they can work together to enforce both network and data security.

We propose to develop a new mechanism for audit based detection of collaborative packet drop attacks. We first study the vulnerability of the REAct system [3] and illustrate that collaborative adversaries can compromise the attacker identification procedure by sharing Bloom filters of packets among them. To defend against such attacks, we propose a new mechanism to generate node behavioral proofs. Every intermediate node needs to conduct only a hash calculation on the received packet. In the new approach, a collaborative attacker cannot generate its node behavioral proofs if an innocent node before it does not receive the data packets correctly. The new approach will allow the system to successfully locate the routing segment in which packet drop attacks are conducted. We also investigate the security of the proposed approach and design mechanisms to further reduce the overhead on the intermediate nodes.

The remainder of the paper is organized as follows. In Section II we review previous research on detecting packet drop attacks and on collaborative IDS. In Section III, we introduce the REAct system and its vulnerability to collaborative attacks. In Section IV, we present the details of the proposed approach. Specifically, we describe the generation of the packet forwarding commitments and behavioral proofs. We investigate the security of the proposed approach and design schemes to reduce overhead. Finally, Section V discusses future work and concludes the paper.

II. Related Work

A. Detecting Packet Drop Attacks

In the self-organized environment of MANETs, wireless nodes are not motivated to consume their energy to help other nodes forward packets. Therefore, several kinds of packet drop attacks such as black-hole [4] and grey-hole [5] have been investigated. Mechanisms to defend against individual attackers can be divided into three groups: audit-based, credit-based, and acknowledgement-based.

The audit-based approaches take advantage of the omni-propagation of wireless signals and use neighbors to monitor the behaviors of a wireless node. In [1], authors propose two methods, namely watchdog and pathrater, to verify packet forwarding and assess quality of routes. Buchegger and Boudec [6] develop a method to distribute the monitoring results to other nodes in the network. In [7] and [8], both first-hand and second-hand evidences are used to detect misbehaving nodes. The factors that prevent the wide adoption of these approaches are three folds. First, eavesdropping on the network traffic may consume as much as 50% of data transmission energy. Second, by using directional antennas or controlling data transmission power, the attackers can cheat their neighbors with fake data forwarding. Finally, mechanisms must be designed to guarantee the authenticity of the monitoring reports.

Several approaches have been designed to provide incentives to wireless nodes so that they will forward packets for other entities. In [9], wireless nodes will use "nuggets" to represent credits for packet forwarding. The approach depends on tamper-proof hardware to guarantee that the credit number will not be changed by unauthorized entities. In [10] and [11], the wireless nodes depend on a centralized server or a base station to manage their credits. These approaches are usually proactive methods and may cause large overhead during the routine operations of MANETs.

To prove that a wireless node has actually forwarded packets to the next hop, the receiver can send acknowledgements in the reverse direction for multiple hops. Two-hop acknowledgements are sent in [12] to achieve the goal. In [13], pilot packets that cannot be distinguished from real data packets are sent to evaluate the routes. Similar to the credit-based approaches, these schemes are also proactive methods and will incur extra communication overhead on the wireless nodes. At the same time, special methods for key management must be designed for the authenticity of the acknowledgements.

B. Collaborative Attacks and Detection in MANETs

Researchers have noticed the threat of collaborative attacks on MANETs and designed several mechanisms to defend against them. In [14], the author provides a proper definition and categorization of collaborative attacks against MANETs from various multiple node attacks found. Specifically, the author investigates the performance impacts of a collaborative blackhole attack on a mobile ad hoc network and studies several mitigation methods. A collusion attack model against optimized link state routing (OLSR) protocol is presented in [15]. The authors also design a technique to detect the attack by utilizing information of two hop neighbors. Collusive attacks on key management and updates in wireless networks have also be studied [16].

For prevention and detection mechanisms, collaborative intrusion detection systems for MANETs have been designed in [17]. The authors assume a clique or a cluster network structure. Therefore, it is not easy to generalize the methods to large scale, multi-hop MANETs. An honesty-rate IDS [18] makes collaborative decisions based on multiple threshold values including rewards and penalties for packet forwarding. Researchers have also integrated ideas from immune systems to achieve collaborative detection of adversaries [19].

In [22], the authors propose a mechanism to detect Byzantine behaviors during packet forwarding in MANETs. Using the acknowledgements from the destination, the source can find changes in packet delivery. Then a binary search based query procedure is adopted to locate the faulty link in the path. The method can detect both individual and collusive Byzantine behaviors.

III. Collaborative Attack on Audit Based Node Misbehavior Detection

In this section, we investigate the collaborative attack on the REAct system [3] that is a random audit based detector of packet drop attacks. We first present a short introduction of the REAct system. Collaborative attacks to compromise the node behavioral proofs are then discussed.

A. Introduction to REAct System

The REAct system tries to identify individual misbehaving nodes in MANETs that refuse to forward packets because of selfishness or maliciousness. The system assumes that there are at least two node disjoint paths between any pair of nodes in the network. The source knows the identity of every intermediate node on the path and a pairwise key can be used to protect the communication between the source and an intermediate node.

Without losing generality, we assume that there are k intermediate nodes (n_1 to n_k) on the path between S and D. As a reactive method, when the destination D detects a significant drop in packet delivery ratio, it will send feedback to the source S. S will select a node n_i to verify that it correctly receives the packets from the previous hop. To achieve this goal, S will send an audit request to n_i through a path that is different from ($S, n_1, n_2, ---, n_{i-1}, n_i$). The request identifies a group of packet sequence numbers and asks n_i to generate a behavioral proof based on the contents of these packets.

To generate the behavioral proof, n_i will construct a Bloom filter based on the contents of these packets. Since a Bloom filter is much smaller than the total length of the selected packets, the approach will not cause large storage and communication overhead on the audited nodes. After generating the proof, n_i will sign the result and send it to S.

The source node S will also generate its own Bloom filter based on the selected packets. When S receives the behavioral proof from n_i, it will compare the two vectors. If the two filters are similar, S concludes that the misbehaving node is in the

path segment between n_i and D. Otherwise, the misbehaving node is in the segment from S to n_i. The source node will then select the next audited node from the smaller segment. This procedure will continue until only two neighboring nodes are left in the suspicious set. The link will then be removed from the path and a new route will be detected. Fig. 1 illustrates an example of the proposed approach. S will first select n_4 as the audited node. Since n_4 can successfully generate the proof, S concludes that the attacker is in the segment from n_4 to D. This procedure will repeat until the link of n_5 and n_6 is located and removed from the path.

If the REAct system adopts binary search to locate the misbehaving node, the attacker can easily predict the order in which the nodes are audited. Therefore, it can dynamically change its behavior to cheat the source. To mitigate such attacks, REAct uses random binary search. More details of the methods can be found in [3].

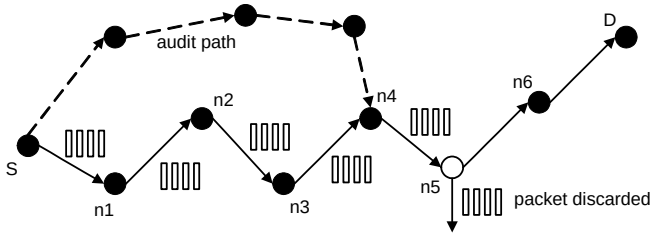

Figure 1. S selects n_4 to be the first audited node.

B. Collaborative Attack on REAct

The REAct system is designed to detect individual misbehaving nodes. Therefore, the assumption of the approach is that a node can successfully generate the behavioral proof only when it receives all selected packets. This assumption, however, will no longer hold when the adversaries work together. Fig. 2 illustrates an example.

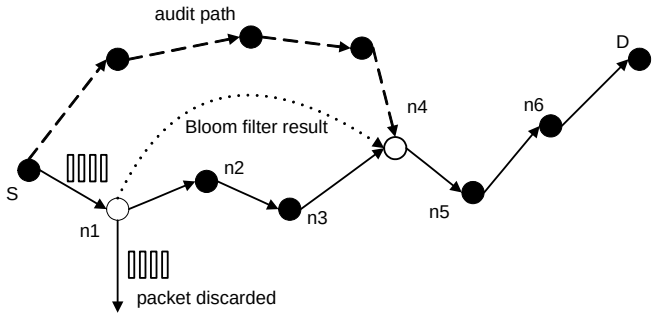

Figure 2. Collaborative attacks on random audit.

In the path between S and D, n_1 and n_4 are two attackers that can communicate to each other through a side channel. The node n_1 discards every data packet passing through it. When S selects n_4 to be the audited node, it will send n_1 the sequence numbers of the selected packets. n_1 will construct the Bloom filter of these packets before discarding them. The Bloom filter will then be sent to n_4, which will be forwarded to S. In this way, the attackers successfully lead the focus of the detection algorithm to the wrong segment of the path. To make the scenario even more complicated, if the source S audits n_1, n_3, and n_4, it will get conflicting behavioral proofs. While both n_1 and n_4 pass the detection procedure, n_3 fails to generate the Bloom filter. The source will not be able to identify the adversary based on the conflicting results.

The main reason that REAct is vulnerable to collaborative attacks is that the Bloom filter based node behavioral proof contains only information from the packets but not from the forwarding path. Therefore, the source node cannot verify which node on the path generates the proof. To solve this problem, in the next section we will present a new method to generate node behavioral proofs using only hash functions. The new approach will cause very limited overhead on the intermediate nodes.

IV. Proposed Approach

In this section we present the details of the proposed approach. We first describe the assumptions of the system. The new generation procedure of the behavioral proofs will then be presented. Finally, we study the safety of the proposed approach and discuss schemes to further reduce its overhead.

A. System Assumptions

We adopt a system model that is very similar to that of the REAct approach. We assume that the source knows the identity of every intermediate node on its path to D. This can be achieved through the adoption of a source routing protocol such as DSR [23]. There exist at least two node disjoint paths between any pair of nodes. We also assume that the source S shares a different symmetric key k_i and a random number r_i with every intermediate node n_i [24]. S and the intermediate nodes have agreed on a secure hash function $h()$. When there is a significant performance drop in the packet delivery ratio, the destination will send an alarm to the source to trigger the audit procedure.

We assume that there are multiple malicious nodes in the network and they may appear in the path between S and D. We assume that the attackers will share their secrets and they have a side channel to communicate with each other. Therefore, a malicious node can impersonate any other attackers in the group. The attacker will drop the data packets passing through it and other adversaries will generate fake information to help it avoid detection.

B. Hash Based Node Behavioral Proofs

The proposed approach works in the similar way as the REAct system except for the generation of the node behavioral proofs. When the source node S determines the audited node n_i, it will send the sequence numbers of the selected packets to n_i. When S sends out these packets, a newly generated random number will be attached to the end of each packet. Therefore, the format of the sent packet is as follows:

$$S \rightarrow n_1\text{: } (S, D, \text{data packet, random number } t_0) \quad (1)$$

Node n_1 will combine the received packet and its random number r_1 to calculate the value t_1 and attach it to the packet when it forwards the data.

$$t_1 = \text{h}(r_1 \,||\, S \,||\, D \,||\, \text{data packet} \,||\, t_0 \,||\, r_1) \quad (2)$$

$n_1 \rightarrow n_2$: (S, D, data packet, t_1) (3)

Here "||" represents the concatenation operation. The intermediate node uses its random number to "sandwich" the received packet and calculate the new commitment of the packet and the forwarding path. This procedure will continue until n_i receives the packet.

When n_i receives the packet, it will first calculate the value of t_i using Equation (2). It will then feed the received data packet and t_i to the Bloom filter to update the node behavioral proof. The audited node will continue these operations until all packets selected by S have been received and the behavioral proof has been generated. It will then encrypt the proof with the key k_i and send it back to the source.

S will verify the correctness of the node behavioral proof when it receives the data. Since it has the knowledge of the data packets and the random numbers t_0 and r_1 to r_i, the source node can reconstruct the commitments of the packets and generate its own copy of the Bloom filter. It will then compare this value to the received behavioral proof. If the difference between the two vectors is smaller than a threshold, S will conclude that the misbehaving node is in the segment from n_i to D. Otherwise, the attacker is in the segment from S to n_i. The source will then select the next audited node from the updated suspicious set.

The node behavioral proofs in our proposed approach contain information from both the data packets and the intermediate nodes. The following analysis shows that this method can defend against the collaborative attacks discussed in Section III.B.

Theorem 1. If node n_i correctly generates the value t_i, then all innocent nodes in the path before n_i (including n_i) must have correctly received the data packet selected by S.

Proof: We prove this theorem by contradiction. Without losing generality, we assume that there exists an innocent node n_j on the path between S and D, and we have $j < i$. We assume that node n_j does not receive the correct data packet. Therefore, it has a very high probability to generate a hash result that is different from the correct value of t_j. On the other side, since node n_i generates the right value of t_i, it must have received the correct value of t_{i-1}. We can repeatedly apply this derivation and conclude that node n_{j+1} must receive the correct value of t_j from node n_j. Since we already know that node n_j calculates the wrong value of t_j, we find the contradiction.

We can apply the same procedure to prove the theorem when $j = i$. ■

In the proposed approach, the behavioral proof contains not only the information about the data packets but also the history of the forwarding nodes. The ordered hash calculations guarantee that any update, insertion, and deletion operations to sequence of forwarding nodes will be detected. With this theorem proven, we can show that the new approach will help wireless nodes defend against collusive attacks described in Section III. When the source node selects to audit node n_i, the returned behavioral proof will determine its next operation.

1) if the behavioral proof passes the test of S, the suspicious set will be reduced to {n_i, n_{i+1}, ---, D*}:*

If the node n_i is innocent, based on theorem 1, we know that n_i must have correctly received the packets selected by S. Therefore, there are no misbehaving nodes from S to n_{i-1} for these packets.

If the node n_i is malicious, based on theorem 1 we know that the closest innocent node n_j before n_i must correctly receive and forward the packets. Therefore, all innocent nodes before n_i have been removed from the suspicious set. n_i as a malicious node is still in the suspicious set and its behavior will be monitored.

2) if the behavioral proof fails the test of S, the suspicious set will be reduced to {S, n_1, ---, n_i*}:*

Since n_i generates the wrong behavioral proof, some node from S to n_i must have received the wrong data packets. The source reduces the suspicious set to the right targets.

Under both conditions, the proposed approach will generate the correct suspicious set for following detections. Using the methods described in [3], the source will be able to locate the attacker that drops the packets continuously or following a sophisticated pattern.

C. Discussion

When a security mechanism is designed to improve an existing approach, we must investigate the safety of the scheme and its overhead. Below we study these problems.

Indistinguishable Audit Packets

If an attacker can distinguish audit packets from common data packets, it will adjust the misbehavior to avoid detection. Therefore, the proposed approach will lead to the following changes to the data packet format in the network. When the source node sends out a data packet, it will attach a newly generated random number to the end of the packet. All intermediate nodes will calculate the commitments of the packet and forwarding path when they receive it. Based on the sequence number of the packet, an audited node will determine whether or not to add it into its Bloom filter. Other nodes, however, cannot tell the difference between an audit packet and a common data packet.

Attaching extra information to data packets will introduce new communication and computation overhead on intermediate nodes. Different applications may choose the length of the commitments based on their security requirements. We believe a 128-bit hash result is good enough for the proposed approach since every intermediate node uses its own secret to calculate the hash result. The probability that two data packets having the same hash results at all intermediate nodes will decrease exponentially as the path length increases. With this configuration, an intermediate node needs to send sixteen more bytes for every data packet. Mechanisms to reduce the computation overhead will be discussed later.

Reducing Computation Overhead

Previous research shows that a hash function needs about 20 machine cycles to process one byte [20]. To reduce the

computation overhead on the intermediate nodes, we propose to allow them to use a part of the data packets to generate the commitments. Below we describe the details of the method.

We assume that the source node *S* and an intermediate node n_i can use their shared secret r_i and a public function *f()* to jointly select *m* bytes from the data packet. Now the commitment of n_i will become:

$$t_i = \mathrm{h}(\, r_i \,||\, S \,||\, D \,||\, m \text{ bytes from data packet} \,||\, t_{i-1} \,||\, r_i \,) \quad (4)$$

The system can control the computation overhead on the intermediate nodes by adjust the value of *m*. If *m* equals to 10% of the packet length, we can avoid the majority of the computation. The probability that an attacker randomly chooses *m* bytes from the packet and they have the same value and order as the outputs of *f()* is fairly low when *m* is reasonably large. The probability that all commitments are correct will decrease exponentially as the number of intermediate nodes increases. Therefore, this improvement will not hurt the safety of the approach badly.

Security of the Proposed Approach

In this part, we investigate the safety of the proposed approach. Since the method uses only hash functions to generate the commitments of the data packets and previous research shows that even mobile devices can conduct this operation very efficiently [20], it will be very difficult to conduct Denial-of-Service attacks on the proposed approach. The collaborative attackers may try to generate fake commitments of innocent nodes. Following the proof in [21], we can show that the adversaries have to have a non-negligible advantage in breaking the hash function to accomplish this task. Therefore, the proposed approach is robust against the attack if the hash function is considered safe.

In collaborative attacks, when an adversary receives the audit request, it will notify other attackers to adjust their behaviors to avoid detection. To improve the detection success rate of the approach, we plan to adopt two methods. First, the source *S* can ask several nodes to generate the behavioral proofs using the same group of packets. In this way, the source node can cross-reference multiple proofs to locate the misbehaving nodes. At the same time, using the same group of packets to monitor multiple nodes will help to reduce the detection delay. Second, the source should adopt a random pattern to select the nodes under audits. In this way, an attacker will not be able to predict the suspicious set based on the value of its behavioral proof. By randomly generating the nodes under audits, the source can get multiple overlapping suspicious sets. It can then use a voting algorithm to locate the misbehaving link.

V. CONCLUSIONS

In this paper, we propose a new mechanism for wireless nodes in MANETs to generate behavioral proofs for the detection of packet drop attacks. Our analysis shows that previous approaches are vulnerable to collaborative attacks. We design a hash based method to generate packet commitments that contain information from both data traffic and forwarding paths. The new method is robust against the collaborative attacks discussed in the paper and it introduces limited computational overhead on the intermediate nodes. We also investigate the security of the proposed approach and design schemes to further improve its detection efficiency and reduce the overhead.

Immediate extensions to our approach include the following aspects. First, we plan to investigate other collaborative attacks on MANETs and design new mechanisms to detect them. Second, we plan to integrate the proposed approach with other methods such as secure routing protocols to construct a comprehensive scheme to protect mobile ad hoc networks. The research will promote the adoption of MANETs by future applications.

REFERENCES

[1] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proceedings of the 6th Annual international Conference on Mobile Computing and Networking (MobiCom), pp. 255-265, 2000.

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

[3] W. Kozma, and L. Lazos, "REAct: resource-efficient accountability for nodemisbehavior in ad hoc networks based on random audits," in Proceedings of the Second ACM Conference on Wireless Network Security (WiSec), pp. 103-110, 2009.

[4] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Wireless Networks, 11(1):21–38, 2005.

[5] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis, "Secure Routing and Intrusion Detection in Ad Hoc Networks," in IEEE International Conference on Pervasive Computing and Communications, pp. 8–12, 2005.

[6] S. Buchegger and J.-Y. L. Boudec, "Self-policing mobile ad-hoc networks by reputation systems," IEEE Communications Magazine, pp. 101-107, 2005.

[7] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputation-based incentive scheme for ad hoc networks," in IEEE WCNC, 2004.

[8] P. Michiardi, and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in Proceedings of IFIP Joint Working Conference on Communications and Multimedia Security, pp.107-121, 2002.

[9] L. Buttyán, and J. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," Mobile Networks and Applications, 8(5), pp. 579-592, 2003.

[10] M. Jakobsson, J.-P. Hubaux, and L. Buttyan, "A micropayment scheme encouraging collaboration in multi-hop cellular networks," in Financial Crypto, 2003.

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

[12] K. Liu, J. Deng, P. Varshney, K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs," IEEE Transactions on Mobile Computing, 6(5), pp. 536-550, 2007.

[13] V. Padmanabhan, D. Simon, "Secure traceroute to detect faulty or malicious routing," ACM SIGCOMM Computer Communication Review, 33(1), pp. 77-82, 2003.

[14] Cong Hoan Vu, Adeyinka Soneye, "An Analysis of Collaborative Attacks on Mobile Ad hoc Networks," Master Thesis at School of Computing, Blekinge Institute of Technology, 2009.

[15] B. Kannhavong, H. Nakayama, A. Jamalipour, "A Collusion Attack Against OLSR-based Mobile Ad Hoc Networks," in IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-5, 2006.

[16] M. Younis, K. Ghumman, M. Eltoweissy, “Key management in wireless ad hoc networks: collusion analysis and prevention,” in IEEE Performance, Computing, and Communications Conference (IPCCC), pp. 199- 203, 2005.

[17] N. Marchang, and R. Datta, “Collaborative techniques for intrusion detection in mobile ad-hoc networks,” Ad Hoc Netw. 6(4), pp. 508-523, 2008.

[18] P. Sen, N. Chaki, R. Chaki, “HIDS: Honesty-Rate Based Collaborative Intrusion Detection System for Mobile Ad-Hoc Networks,” Computer Information Systems and Industrial Management Applications (CISIM), pp.121-126, 2008.

[19] K. Yeom and J. Park, “An immune system inspired approach of collaborative intrusion detection system using mobile agents in wireless ad hoc networks”, in International conference of Computational intelligence and security, 2005.

[20] B. Preneel, *et al*, “Performance of optimized implementations of the nessie primitives,” Deliverable 21 from the NESSIE IST FP5 project, 2003.

[21] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, “Dynamic and efficient key management for access hierarchies,” ACM Trans. Inf. Syst. Secur., 12(3):1–43, 2009.

[22] 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. 10(4), 1-35, 2008.

[23] David B. Johnson, “Routing in Ad Hoc Networks of Mobile Hosts,” in Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 158-163, 1994.

[24] M. Khatib, K. Masmoudi, and H. Afifi, “An on-demand key establishment protocol for MANETs,” International Conference on Advanced Information Networking and Applications (AINA), 2006.