An Efficient Black Hole Detection Method using an Encrypted Verification Message in Mobile Ad Hoc Networks

Firoz Ahmed, Seok Hoon Yoon and Hoon Oh
University of Ulsan
jewelraaz@yahoo.com, {seokhoonyoon, hoonoh}@ulsan.ac.kr

Abstract

We propose an Encrypted Verification Method (EVM) that effectively detects a black hole attack. A detection node that receives an RREP from a suspicious node sends an encrypted verification message directly to destination along the path included in the RREP for verification. The approach not only pins down the black hole nodes, but also reduces control overhead significantly. We prove by resorting to simulation that EVM is highly dependable against the black hole attack.

Keywords: *Black hole attack, encryption, decryption, AODV, MANET*

1. Introduction

Mobile ad hoc networks (MANETs) are formed autonomously by a number of mobile nodes (also act as a router) without the help of a centralized management entity. Since a mobile node can join or leave the network without permission from a management entity, MANETs are vulnerable to various kinds of attacks such as black hole attack [1], worm hole attack [2] and so on. We address the black hole attack problem when AODV [3] is used for routing in MANETs.

A malicious node sends an RREP with a high destination sequence number in response to an RREQ in order to intercept data packets destined for destination. This is called a black hole attack. It can be categorized into two types: A single black hole attack and a colluding black hole attack. In a single black hole attack, a malicious node act alone whereas in colluding black hole attack, two or more malicious nodes that collaborates each other to deceive the other nodes more effectively.

The single black hole attack has been tackled in various ways. Some focused on verifying the correctness of the obtained path through the downstream node of the RREP initiator [1], [4]. In [5], a watchdog mechanism has been used that a node watches the misbehavior of its downstream node. However, these approaches may not work appropriately if two black hole nodes cooperate with each other. Meanwhile, only a few methods have been proposed to tackle the colluding black hole attack. In SNV [6], every RREP initiator sends a message to destination to ask for the destination to report its current sequence number to the source. However, it produces high control overhead and makes false decision to detect malicious node in relatively high mobility networks.

The approaches discussed above suffer from high overhead by using flooding or additional messages as well as the failure to address colluding attack. An encrypted verification method (EVM) proposed in this paper can resolve this problem effectively through two step approaches: Identification of a suspicious node and the verification of the suspicious node using an encrypted verification message. Thus, the malicious behaviors of a node on the path such as the fabrication, dropping or absorption of the message can be detected effectively. Simulation results show that the EVM can reduce control overhead and increase the detection rate considerably compared to the SNV.

The rest of the paper is organized as follows. The proposed method is described formally in Section 2. The performance evaluation on the effect of EVM is given in Section 3 and is followed by concluding remarks in Section 4.

2. Encrypted Verification Method (EVM)

2.1. Identification of Suspicious Node

For identifying a suspicious node, each node collects data by overhearing the packets that its neighbors transmit and maintains a data collection table (*DCT*) with those data as follows.

$$DCT_i = (j, From_j, Through_j, Suspecious_j),\ j \in i.N, \text{ where}$$

- $i.N$ is a collection of node *i's* neighbors;
- $From_j$ indicates whether or not node i has received a packet from node j ever;
- $Through_j$ indicates whether or not node i has routed a packet via node j ever; and
- $Suspicious_j$ indicates whether or not node j is suspicious based on the combination of $From_j$ and $Through_j$ fields.

The values of $From_j$, $Through_j$, and $Suspicious_j$ are given true (1), false (0), non-decidable (x).

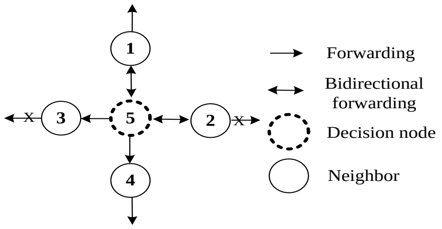


Figure 1. An Example Topology to Define *DCT*

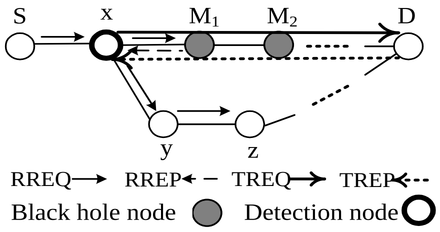
Table 1. An Example of DCT_5

j	$From_j$	$Trough_j$	$Suspicious_j$
1	1	1	0
2	1	0	0
3	0	0	1
4	0	1	x

Take a look at Figure 1 and Table 1. Node 5 observes the data forwarding behaviors of its neighbors and records them in its data collection table, DCT_5. Node 5 has received data packet from nodes 1 and 2 ($From_1$ = $From_2$ = 1). Thus it determines that both are reliable ($Suspecious_1$ = $Suspecious_2$ = 0). However, node 3 did not send data to anyone, including node 5 and is determined to be suspicious ($Suspecious_3$ = 1). As for node 4, node 5 cannot know whether node 4 has forwarded to a reliable node or a malicious node, and thus determines node 4 to be non-decidable ($Suspecious_4$ = x).

For the non-decidable node, we need further observation. It may be reasonable to assume that multiple different paths can go through the non-decidable node. A decision node can count the number of different downstream nodes to which the non-decidable node forward data packet by using the watchdog mechanism. If the number is over some threshold, it can determine the non-decidable node to be reliable.

2.2. Encrypted Verification Process



1. *S* broadcasts RREQ to explore a path
2. M_1 responds with the fake RREP
3. If x determines M_1 to be suspicious, it sends the encrypted TREQ along the path specified in RREP
4. *D* receives the TREQ and responds with the encrypted TREP
5. *x* compares the sequence number in RREP and TREP to verify M_1

Figure 2. Verification Process

If a source or an intermediate node receives RREP from a reliable node, it takes the exactly the same process as AODV. That is, the source starts sending data packets while the intermediate node forwards the RREP toward the source. A node that receives RREP from a suspicious node, initiates a verification process to check if the suspicious node is a black hole.

The node (detection node) extracts destination sequence number (dsn_1) from the RREP and stores it in its cache. It then generates a Test Request message, TREQ = (*detection node address*, *destination node address*, *timestamp*) where the *timestamp* indicates a current time. The detection node encrypts TREQ using public key cryptosystem and sends it along the path specified in the RREP towards destination. A node that receives the TREQ relays it to next node. Upon receiving the TREQ, the destination node decrypts the message and creates a Test Reply message, TREP = (*detection node address, destination node address, timestamp,* dsn_2) where *dsn* indicate current destination sequence number. The destination encrypts the TREP and sends it along the reverse path to the detection node. Upon receiving the TREP, the detection node decrypts it. If $dsn_1 >> dsn_2$, the detection node judges the suspicious node is black hole and drops the RREP. If the suspicious node is determined to be a reliable node, the detection node starts sending data packets to the destination if the detection node is source; otherwise, it forwards the RREP towards the source node. If detection node does not receive TREP until the timer expires, it refuses to forward the RREP to its upstream node. Then, the source will consider other paths contained in some other RREPs that it has received.

Figure 2 illustrates a verification process. When a suspicious node M_1 responds with RREP, detection node *x* sends the encrypted TREQ to destination *D* along the path specified in RREP. Upon receiving the TREQ, Node *D* responds with the encrypted TREP along the reverse path towards *x*. M_1 and M_2 cannot alter the contents of the encrypted RREP. Detection node *x* judges whether M_1 is reliable or not by comparing dsn_1 and dsn_2.

2.2.1. Encryption and Decryption

We assume that every node has a pair of public and private keys that are used in the RSA public key cryptosystem [7]. The public key of each node can be distributed once when it joins in a considered network. The newly joined node can get the public keys of the other nodes in the network from one of its neighbors.

If a node receives RREP from a suspicious node, it (or detection node) extracts destination sequence number and stores it in its cache. The detection node creates TREQ and encrypts the detection node address with the public key of destination node. Now, the detection node forwards the TREQ along the forward path towards destination after it signs the whole TREQ with its own private key. The receiving node validates and removes previous node's signature in the TREQ and checks whether it is the destination or not by checking the destination IP address. If it is not the destination, it takes the same process again. The process continues until the TREQ reaches the intended destination.

Given a forward path = (N_1, N_2 … N_{l-1}, N_l) where N_1 and N_l represent detection node and destination node, respectively. IP_x is the IP address of node x, and K_x+, K_x- is the public and private key of x (here x indicates node id). The behavior of each node on the path can be described formally as follows.

$$N_i \rightarrow N_{i+1} : \lfloor \lfloor IP_{N_1} \rfloor K_{N_l}+, IP_{N_l}, timestamp \rfloor K_{N_i}-, 1 \le i \le l-1$$

Upon receiving TREQ, the destination node creates TREP message in which both the destination node address and the dsn_2 are encrypted with the public key of detection node. Now, the destination node forwards the TREP along the reverse path after it signs the whole TREP with its own private key. The receiving node validates and removes previous node's signature in the TREP and checks whether it is detection node or not by checking the IP address of detection node. If it is not the detection node, it takes the same process again. The process continues until the TREP reaches the detection node. The behavior of each node can be described formally as follows.

$$N_{l-i} \rightarrow N_{l-i-1} : \lfloor \lfloor IP_{N_l}, dsn_2 \rfloor K_{N_1}+, IP_{N_1}, timestamp \rfloor K_{N_{l-i}}-, 0 \le i \le l-1$$

3. Performance Evaluation

Using the NS-2 [8], we compare EVM and SNV with Random Waypoint Model. The used simulation parameters are given in Table 2. The simulation for each scenario was performed five times and then the average value for each metric was presented. We use four metrics: Packet delivery rate, control overhead, true positive rate and false positive rate.

Table 2. Simulation Parameters

Parameter	Value	Parameter	Value
Number of nodes	50	Number of sessions	15
Terrain range	1000 * 1000 m^2	Simulation time	300 sec.
Maximum speeds	0, 5, 10, 15, 20, 25m/s	Packet transmission rate	4 packets/s
Pause time	30s	Number of malicious nodes	1, 2, 3, 4, 5

Figure 3 and Figure 4 show packet delivery rate and control overhead with varying number of malicious nodes, respectively. The packet delivery rate of all schemes performs well in case of no black hole node. A significant result is that the packet delivery rate of AODV dramatically drops from 88 percent to 21 percent in the presence of one black hole node and it becomes worse as the number of black hole nodes increase. EVM has control overhead lower than SNV since it can send an encrypted verification message directly to destination which is protected from the modification of other nodes. EVM can detect a black hole more reliably for the same reason. The control overhead of AODV sharply decreases as the number of malicious nodes increases since a black hole node tends to hinder normal protocol operation such as message forwarding. EVM shows control overhead lower than SNV since it does not use flooding.

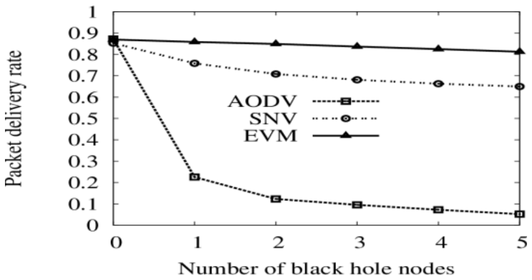


Figure 3. Packet Delivery Rate versus Number of Black Hole Nodes

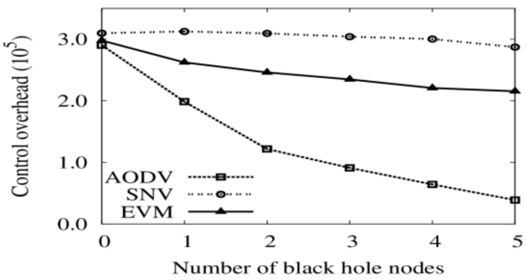


Figure 4. Control Overhead versus Number of Black Hole Nodes

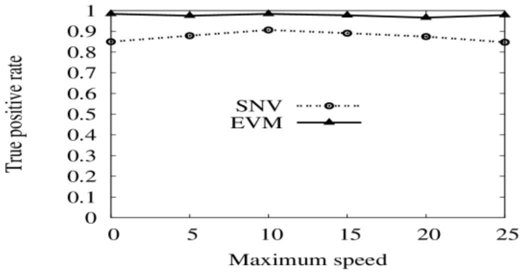


Figure 5. True Positive Rate versus Maximum Speed

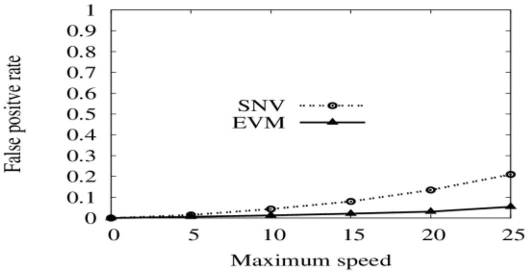


Figure 6. False Positive Rate versus Maximum Speed

Figure 5 and Figure 6 show true positive rate and false positive rate with varying maximum speed and 10% of total nodes are black hole nodes. The true positive rate of SNV is lower than that of EVM because in SNV, only source node decides that a node is malicious. Therefore, if a certain node detects anomaly of a black hole node it floods alarm message that is supposed to be delivered to the source. If the alarm message fails to reach to the source, SNV cannot detect black hole node. The false positive rate in both schemes increases when the nodes move more rapidly because links are broken frequently in a high mobility network. We also observe that the false positive rate of SNV is higher than that of EVM. This is due to the fact that in SNV, if the RREP originator is disconnected from its immediate upstream node or it fails to send SREQ to its downstream node due to a link breakage, the upstream node determines that the RREP originator is black hole.

4. Concluding Remarks

The proposed EVM method can pin down multiple black hole nodes effectively by employing an encryption mechanism. The verification process is initiated conditionally and it verifies the sequence number that was not faked by any malicious node. We show by simulation that the EVM not only reduces the control overhead but also effectively identifies the malicious node. In the future, we will extend our algorithm to solve a selective forwarding attack such as gray hole attack.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (**NRF**) funded by the Ministry of Education, Science and Technology (2011-0004606).

References

[1] H. Deng, W. Li and D. P. Agarwal, "Routing security in wireless ad hoc networks". In: IEEE Communications Magazine, 40, 70–75, **(2002)**.

[2] Y. -C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks". In: Proc. of IEEE INFOCOM **(2002)**.

[3] C. E. Perkins, E. M. Royer and S. Das, "Ad-hoc On demand Distance Vector (AODV) Routing Protocol". In: RFC, pp. 3561, **(2003)**.

[4] S. Lee, B. Han and M. Shin, "Robust Routing in Wireless Ad Hoc Networks". In: International Conference on Parallel Processing Workshops, pp. 18-21, **(2002)**.

[5] S. Marti, T. Giuli, K. Lai and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", In: MOBICOM, pp. 255-265, **(2000)**.

[6] XiaoYang Zhang, Yuji Sekiya and Yasushi Wakahara, "Proposal of a method to detect black hole attack in MANET". In: International Symposium on Autonomous Decentralized Systems, pp. 1-6, **(2009)**.

[7] W. Stallings, "Cryptography and Network Security Principles and Practices", Fourth Edition, Prentice Hall, **(2005)** November 16.

[8] NS-2, http://www.isi.edu/nsnam/ns/.

Authors

Firoz Ahmed is a Ph.D. candidate in the School of Computer Engineering and Information Technology, University of Ulsan, South Korea. He received M.Sc. degree from University of Madras, India, in 2002. He is an assistant professor of department of Information and Communication Engineering (ICE), University of Rajshahi, Rajshahi, Bangladesh. His research interests include security in mobile ad hoc networks and wireless sensor network.

Dr. Yoon is with University of Ulsan, South Korea, as an assistant professor in Computer Engineering and Information Technology. He received MS and PhD degrees in computer science and engineering from the State University of New York (SUNY) at Buffalo in 2005 and 2009, respectively. His research interests include Mobile Sensor/Actuator Networking, Vehicular Networks, and integration of heterogeneous networks.

Hoon Oh is an associate professor of the School of Computer Engineering and Information Technology and a director of the Vehicle IT Convergence Technology Research Center in the University of Ulsan, Korea. He is a member of IEICW, ISCA, KICS, and ICASE, and has been life time member of the Korea Information Society since 1989. His research interests lie in mobile ad hoc networks, real-time computing, and ubiquitous computing.