Mitigating Flooding Attacks in Mobile Ad-hoc Networks Supporting Anonymous Communications

Venkat Balakrishnan, Vijay Varadharajan, and Uday Tupakula
Information and Networked Systems Security Research (INSS) Group, Department of Computing, Macquarie University, Sydney, Australia
{venkat, vijay, udaya}@ics.mq.edu.au

Marie Elisabeth Gaup Moe
Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway
marieeli@q2s.ntnu.no

Abstract

Recently several techniques that provide different degree of anonymity have been proposed for wired and wireless communication. Although, the recently proposed techniques are successful in achieving high degree of anonymity, there are some disadvantages associated with the proposed techniques. In this paper we analyze the flooding and packet drop attacks in mobile ad hoc networks that support anonymous communication. Then we propose a novel technique to deal with the flooding attacks. Our approach can efficiently identify and isolate the malicious node that floods the network. In addition, our technique provides a mechanism to identify the benign behavior of an expelled node and rejoins the expelled node back into the network. Furthermore, our approach does not require any additional packets to communicate the behavior of the flooding node and hence does not incur any additional overhead. Finally we validate the performance analysis of our technique through NS2 simulations.

1. Introduction

Some of the mission critical applications such as battlefield require high degree of anonymous communication. Recently several techniques [1-5] have been proposed to achieve anonymous communication in wired and wireless networks. Currently there is an increasing interest in developing protocols that can achieve high degree of anonymity for communications in Mobile Ad hoc NETworks (MANET) [2-5]. The main aim of anonymous communication is to achieve high resistance to eavesdropping and traffic analysis. In general, most of the proposed techniques are based on public key cryptography and/or based on Chaum's [1] Mix technique. The idea of Chaum's Mix-net concept is that the traffic from a source to a destination has to pass through one or more Mixes. A Mix relays data from different end-to-end connections. However, unlike the traditional routers, a Mix reorders and re-encrypts the incoming packets in such a way that the incoming and outgoing packets from the Mix cannot be related. This is to thwart the attempts of an attacker to follow an end-to-end connection.

Although it is possible to achieve reasonable degree of anonymity in wired networks, there are several additional challenges in achieving anonymity in MANET communications. These are due to the nature of the wireless medium of communication, and limited computational capability of the nodes. Furthermore, a Mix-net should be capable of providing anonymity even if some of the Mixes are compromised. The main reason for this requirement is that in a hostile environment such as a battlefield, there is a greater probability for the roaming nodes or Mixes to be captured by the enemies. Hence, the traffic should be passed through a greater number of Mixes to lower the probability of compromising the anonymity. On the other hand, relaying data traffic through too many Mixes will increase the average latency and decrease the average data delivery ratio. This will also consume greater resources of the mobile nodes. So there is a need to achieve balance between achieving high degree of anonymity and low latency.

Some of the recently proposed protocols [2-5] are successful in achieving different degree of anonymity with reasonable level of latency. However there are some significant disadvantages associated with these approaches. First they are inefficient in terms of performance. Second, they make it extremely

The 2nd International Conference on Wireless
Broadband and Ultra Wideband Communications (AusWireless 2007)
0-7695-2842-2/07 $25.00 © 2007 IEEE

straightforward for an attacker to perform different types of attacks such as flooding and packet drop attacks within the network. Currently available security tools [6, 7] which can detect these abnormalities cannot be used in the case of anonymous communication. We will discuss these issues more in detail in Section 3.

In this paper, we will consider anonymous secure routing (ASR) protocol [2] and analyze how an attacker can severely degrade the performance of the network by performing flooding and packet drop attacks. We will then propose a novel technique to deal with the flooding attacks. The main reason for choosing the ASR protocol is that this protocol is simple, light weight and provides higher degree of identity anonymity, location privacy, and route anonymity, while ensuring the security of established routes in MANET.

Although we analyze packet drop attacks in this paper, we confine our proposal to only flooding attacks. This is because, in order to deal with packet drop attacks, the proposed technique should be able to relate the incoming and outgoing packets from a node. This contradicts with the requirements of anonymity service.

The paper is organized as follows. In Section 2, we give an overview of the ASR protocol for MANET. In Section 3, we analyze how the attacker can perform flooding and packet drop attacks and why it is difficult to deal with these attacks. In Section 4, we present our technique to deal with flooding attacks. Section 5 then analyses the performance of our technique through NS2 simulations. Finally, Section 6 gives some concluding remarks.

2. Anonymous Secure Routing Protocol

The route establishment process of the ASR [2] protocol is similar to Dynamic Source Routing (DSR) protocol [8]. The main difference between the ASR and the DSR protocol is that the route request in the ASR protocol neither contains the source node's address nor the destination node's address. In addition, the intermediary nodes do not append their identity to the route request. In the following, we only present how the multi-hop anonymous routes are established between the source and destination. For more detail on the data transmission, route maintenance and the analysis on the degree of anonymity, we encourage the reader to refer to [2].

The authors of ASR protocol assume that there exist a shared secret between the source and destination. Let us consider how anonymous multi-hop routes are established between the source S and destination D through intermediary nodes X_n. The format of the route request is listed below:

$$\left[\text{RREQ, seq, } K_T(\text{dest}, K_s, U_o),\ K_s(\text{seq, END}),\ PK_{i-1},\ U_{i-1}\right]$$

'RREQ' specifies that it is a route request, 'seq' is the sequence number, 'K_T' is the shared secret between the source 'S' and destination 'D', 'K_s' is the session key, 'END' is included to denote that the destination has received the route request, 'PK_{i-1}' is the one time public key that is generated by the previous node 'X_{i-1}', 'PK_o' is the one time public key of the source node 'S', 'U_o' is a random number chosen by source 'S' and 'U_{i-1}' is a random number computed by 'X_{i-1}'.

On receiving the route request, each node checks for the sequence number (seq). If the sequence number (seq) is already recorded, then the route request is dropped and no action is initiated. Alternatively, if it is a latest sequence number (seq), then the node records the 'seq' and attempts to decrypt the third element of the route request, i.e., 'K_T(dest, K_s, U_o)'. Given that the decryption is successful, the node then concludes that the route request is destined for it. The route request is further processed to extract the 'U_o' and also for the validation of the max number of hops.

On unsuccessful decryption of the element, 'K_T(dest, K_s, U_o)', the node records 'seq', 'PK_{i-1}', and 'K_s(seq, END)' into its route table. It then generates 'U_i' based on a standard function and replaces 'PK_{i-1}' and 'U_{i-1}' with 'PK_i' and 'U_i', and broadcasts the route request. The process is repeated until the route request reaches the destination.

Similar technique can be used to send the route reply. However the authors have proposed a novel approach to send the route reply in order to identify and eliminate the illegitimate route replies. In this approach, the route replies are forwarded, only if -- the node has previously forwarded the route request, and can ensure that the destination has successfully received the route request. The format of the route reply is listed below:

$$\left[\text{RREP, } \{T_{i+1}\}_{PKi},\ T_{i+1}(\text{seq}, K'_s)\ \right]$$

'K'_S' is the proof that destination has recovered the secret from the third element of the route request. This can be verified by validating the element 'K'_S(seq, END)' stored in the routing table. 'T_{i+1}' is a random number chosen by 'X_{i+1}' and is used as secret between 'X_i' and 'X_{i+1}'. After receiving the route reply, each node tries to decrypt '$\{T_{i+1}\}_{PKi}$' in order to recover the final element of the route reply. Since '$\{T_{i+1}\}_{PKi}$' is encrypted with 'PK_i', only 'X_i' can decrypt the packet.

If the received route reply can be validated by the above process, then the node 'X_i' chooses a random number 'T_i', and adds 'T_i' and 'T_{i+1}' into its record with the corresponding sequence number (seq). In succession, it also computes the '$\{T_i\}_{PKi-1}$' and 'T_i(seq, K'_S)' and replaces the last two elements of the route reply and then broadcasts the route reply. At the end of the route reply phase, each forwarding node would have established a shared secret with the previous-hop node and the next-hop node for that particular communication between S and D.

After the establishment of anonymous routes, the successive data transmission and route maintenance are achieved through the shared secrets. Tags are used to minimize the computation overhead at the nodes for validating and forwarding the data packets. The nodes just need to validate the tags instead of validating the complete data packets. In the following section, we will analyze how the attacker can perform flooding attacks with the ASR.

3. Analysis

ASR provides high degree of mutual anonymity compared to other techniques. However there are some significant disadvantages with this technique. Although the authors state that the initial validation of the route request is done through symmetric key and requires less computational resources, the attacker can flood the network with malicious route requests. Furthermore the route requests can be destined to a non-existing node. In this case, each generated route request will be forwarded by most of the nodes within the network. Since it is not possible to differentiate the packets that are originating from a particular source node or to identify the packets that are destined to a particular destination node, it is extremely difficult to deal with the flooding attacks. For example, a node receiving a flood of packets from its previous-hop node cannot determine whether it is flooded – by its previous-hop node or by the nodes prior to its previous-hop.

In traditional MANET, where communications are non-anonymous, flooding attacks are detected based on the rate at which packets are generated by the source nodes or received by the destination nodes [9-11]. Since it is not possible to track back the source and destination nodes in an anonymous network, it is extremely hard to apply existing approaches to defend flooding attacks in anonymous network.

Similarly, packet drop attacks are detected in traditional MANET by relating the incoming and outgoing packets from a node [6, 7]. Since all nodes in an anonymous network act as Mixes, it is not possible to relate between the incoming and outgoing packets from each node. So it becomes extremely difficult to determine whether the nodes are successfully forwarding the packets or dropping the packets.

Hence, both flooding and packet drop attacks can severely degrade the performance of the networks that support anonymous communication. In the following section we will propose a novel technique to deal with flooding attacks. We will not consider packet drop attacks in this paper for the above reasons.

4. Our Approach

4.1. Assumptions and Attacker Model

We assume that the nodes communicate using a single shared bi-directional wireless channel and all the transmissions and receptions are omni-directional. All the nodes operate in promiscuous mode and do not use any tamper proof hardware. We refer to nodes that are one-hop away as *neighbors* and the region under a node's transmission range as the node's *environment*. We refer to the bandwidth sharing as the *network service*. We refer to nodes that disrupt the availability of network service with an intention to drain other node's resources or to prevent other node's from accessing the transmission medium as *malicious nodes*. We inherit the assumption in ASR that there is a shared secret between any two nodes to enable a node to authenticate its neighbor.

4.2. Overview

In our model, each incoming packet passes through the *rate-limitation* component before being forwarded (transmitted) to the next-hop neighbor. The rate-limitation component diminishes the flooding attacks based on the condition that each node is obligated to share its bandwidth with the neighbors in its environment. To achieve this, the rate-limitation at every node uses a *threshold-tuple*, which is a list of thresholds as given in Table 1.

At the initial stages of deployment, each node initializes its thresholds for other nodes. This in turn aids the elimination of an online centralized authority for managing the thresholds. 'α' gives the maximum number of packets a node can transmit in an interval. It is derived from the empirical results taken over the average number of packets transmitted in an interval by the node and the average number of neighbors in its environment. ‘β’ permits the maximum number of times a flooding node can exceed ‘α’ before being black-listed. Note that ‘β’ is introduced to reduce the effect of false positives and hence it has to be low. ‘γ’ denotes the number of consecutive intervals for which the flooding node has to abide within ‘α’, in order for it to be white-listed or redeemed. In our model, ‘γ’ is higher than ‘β’ for the reason that a blacklisted flooding node has to show more commitment for it to be white-listed.

Table 1. Thresholds for Rate-limitation

α	Maximum number of packets permitted to transmit in an interval
β	Maximum number of intervals the flooding node can exceed ‘α’ before being black-listed
γ	Minimum number of consecutive intervals a flooding node has to abide ‘α’ for being white-listed

In the following, we explain the operation of rate-limitation with respect to the reception end of a benign node.

Rate-limitation: *Every participating node is obligated to share the transmission channel equally with the neighbors in its environment. Nodes failing to oblige are excluded from the neighborhood by means of non-contribution.*

Let us consider the operation of the rate limitation component for route request messages. The rate-limitation component at every node monitors each requesting neighbor’s channel usage at regular intervals of time. For ease of explanation, we define the channel usage as the packets generated during an interval. If the packets transmitted by the neighbor exceeds the pre-determined *transmission-threshold (α)* within a given interval, then the subsequent packets are dropped. If the same neighbor exceeds the *transmission-threshold (α)* by *blacklist-threshold (β)* intervals, then the neighbor is believed to be flooding. As a result, the neighbor is blacklisted as a *flooder* and all the packets received from the flooded neighbor are discarded in the future intervals. We follow *blacklist-threshold (β)* intervals before blacklisting a neighbor in order to prevent accidental blacklisting of the neighbor. However the node continues to monitor the behavior of the blacklisted node in the successive intervals. In order to be white-listed or redeemed, the blacklisted node has to exhibit benign behavior for ‘γ’ intervals, which is known as *whitelist-threshold*. Given the blacklisted neighbor is observed to be benign, the monitoring node then whitelists the neighbor and begins to forward the packets for the neighbor. This not only provides a mechanism for the blacklisted neighbor to rejoin the network, but also enforces the blacklisted neighbor to drain its energy to prove that it has been repenting. Choosing ‘γ’ greater than ‘β’ confirms that the blacklisted neighbor has to repent for its past flooding behavior.

Note that the functionality of rate-limitation is aggregated at all monitoring nodes against a flooding node in an environment. This assures that the flooding node’s impact is confined to a single hop. In addition, it not only eliminates the need for techniques to trace back to the source of flooding node but also drains the resources of the flooding node.

We identify two design options to choose an optimum value for the *transmission-threshold (α)*. The first option is static and is determined during the initial stages of the network. The value of ‘α’ at each node is derived from the operational bandwidth of the transmission medium and the expected average node density per neighborhood. The second option is dynamic and ‘α’ is deduced at each node from the operational bandwidth and total number of its active neighbors. The number of active neighbors is inferred from the recently received requests with *new node identifiers* and failed link-layer acknowledgements. The advantage of the latter design is that the benign node with low density of neighbors can make use of the available bandwidth more efficiently than the earlier design. Otherwise, both the design options operate in the similar way, especially at high node densities. We set the range of *transmission-threshold (α)* as: *$1 \leq MinTans \leq \alpha \leq MaxTrans \leq Bandwidth$.* ‘MinTrans’ corresponds to the minimum value that ‘α’ can take, which may otherwise equate to one packet and the maximum value of ‘α’, i.e. ‘MaxTrans’, can be at most equivalent to the bandwidth of the channel.

The benign nodes which receive multiple route requests from the neighbors can deploy congestion control techniques such as drop trail. Even if the route request packets are dropped at some nodes due to congestions, the routes can still be established through another node, if a valid path exists. This is achievable due to the broadcast nature of the route request. Alternatively, if no paths are available, then the source can detect the failure if it does not receive a route reply within a specified time. In this case the source can randomly wait for a time interval and send another route request.

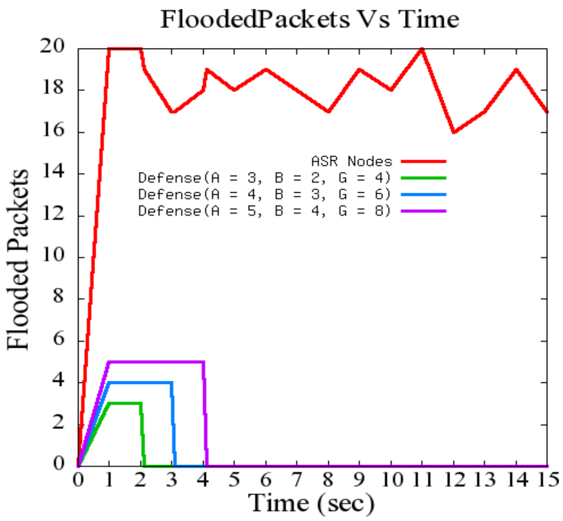

Figure 1: Continuously Flooding Nodes

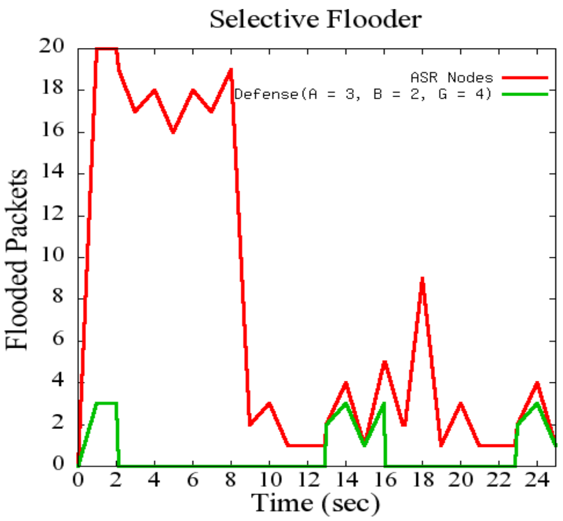

Figure 2: Selectively Flooding Nodes

There are several advantages with our technique. First the proposed technique is very simple. Second the nodes have to maintain only the transmission rates of the neighboring nodes. Hence the computation overhead is very minimal. Furthermore, our technique deals with the flooding attacks *nearest to the source* of attack. This will not only help to save considerable resources for non-flooding mobile nodes, but also enable it to efficiently identify and isolate the flooding node from the network. Even if two nodes are colluding within an environment, the maximum number of packets that can be generated by each colluding node is limited by ‘α’. As each of them exceeds ‘α’ by ‘β’ intervals, the proposed technique restricts their transmissions from being propagated. In addition, our technique provides a mechanism that allows the repenting flooding nodes to rejoin the network, if they exhibit benign behavior for the specified time interval. However, the malicious node is significantly penalized for its past malicious behavior before it can rejoin the network.

5. Simulation Results

We have tested the performance analysis of our model using NS-2 simulator. We summarize the parameters used in the simulation in Table 2.

The mobile nodes which do not enable our model are called *ASR nodes*. The mobile nodes with our model enabled are known as *defence nodes*. The nodes which perform flooding attacks are called as *flooding nodes*.

The total number of nodes for both the combinations – defence and flooding nodes, and ASR and flooding nodes are fixed to 100. Defence and ASR nodes are independently simulated against the flooding nodes under different scenarios. However, the scenarios are kept identical with same set of parameters for comparison purposes. For defence nodes, the simulations are performed for different values of ‘α’, ‘β’ and ‘γ’. The simulation scenarios considered for performance evaluation are given below:

Scenario I - This scenario analyzes the performance of defence and ASR nodes against persistent flooding nodes. Given that maximum 20 packets can be transmitted in an interval, the performance of defence nodes is then analysed for varying thresholds.

Scenario II - This follows the above scenario except it analyzes the performance of defence and ASR nodes against selectively flooding nodes.

Figure 1 shows the graph for continuous flooding attacks. In the graphs ‘α’ is represented as ‘A’, ‘β’ is represented as ‘B’ and ‘γ’ is represented as ‘G’. From the graph it is evident that greater the value of ‘β’, then higher the flooded packets. After interval ‘β’, the malicious nodes are penalized for the period ‘γ’. During this interval, the nodes are monitored for their flooding behavior. Since the nodes are continuously flooding they are identified as persistent malicious

Table 2. Simulation Parameters

Total number of nodes	10
Maximum velocity (V_{max})	20 m/s
Pause time	10s
Simulation area	500x500 m^2
Bandwidth	11Mbps
Payload	1000 bytes
Maximum Packets / Second	20

The 2nd International Conference on Wireless
Broadband and Ultra Wideband Communications (AusWireless 2007)
0-7695-2842-2/07 $25.00 © 2007 IEEE

nodes and hence the packets are continuously dropped. So it can be concluded that the value of 'γ' does not have any relevance for continuously flooding nodes. In summary, it is evident that our approach effectively isolates continuously flooding nodes and restricts the flooded packets within the environment.

Figure 2 demonstrates the effect of 'γ' in defending against selectively flooding nodes. Similar to Figure 1, 'γ' is not activated until the flooding nodes reduce their transmissions to 'α'. Once the flooding nodes transmissions are within 'α' for 'γ' consecutive intervals (from 9s to 12s in Figure 2), then packets are forwarded for the flooding nodes (from 13s to 16s in Figure 2). However, if the flooding nodes exceeds 'α' in alternate intervals after being white-listed (at 14s and 16s in Figure 2), then 'β' is immediately activated to defend the flooding. Nevertheless, if the transmissions of flooding nodes are within 'α' in alternate intervals after being black-listed at 16s (from 17s to 21s in Figure 2), then 'γ' remains deactivated. From the analysis, it is evident that selectively flooding nodes are significantly penalised before they could rejoin the network. From the results, we recommend the 'γ' value to be at least twice than the value of 'β'.

6. Conclusion

We have considered anonymous secure routing protocol for mobile ad hoc network and analyzed how the attacker can perform flooding and packet dropping attacks. Then we have proposed a novel technique to deal with the flooding attacks. The main advantages of our technique are that it can efficiently identify and eliminate the nodes that are flooding the network. The flooding attacks are mitigated nearest to the source of attack. Our approach does not incur any additional overhead in terms of additional packets to identify the benign and malicious behavior of the neighboring nodes. Furthermore our technique provides a mechanism that enables the repenting nodes to rejoin the network. The simulation results confirm these characteristics and hence the technique seems to be very promising for counteracting flooding attacks in mobile ad hoc networks supporting anonymity.

7. References

[1] D. Chaum, "Untraceable Electronic Mail Return Address, and Digital Pseudonyms". *ACM Communications*, Vol. 24 (2), pp. 84-88, 1981.

[2] B. Zhu, Z. Wan, M. S. Kankanhalli, and F. Bao, R. H. Deng, "Anonymous Secure Routing in Mobile Ad-hoc Networks". *Proceedings of the 29th IEEE International Conference on Local Computer Networks (LCN 2004)*, Tampa, USA, pp. 102-108, November 2004.

[3] J. Kong and X. Hong, "ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks". *Proceedings of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2003)*, Annapolis, USA, pp. 291-302, 2003.

[4] Y. Zhang, W. Liu, and W. Lou, "Anonymous Communications in Mobile Ad hoc Networks". *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005).* Miami, USA, pp. 1940-1951, March 2005.

[5] S. Seys and B. Preneel, "ARM: Anonymous Routing Protocol for Mobile Ad hoc Networks". *Proceedings of the 20th IEEE International Conference on Advanced Information Networking and Applications (AINA 2006)*, Vienna, Austria, pp. 133- 137, April 2006.

[6] S. Zhong, J. Chen and Y. R. Yang, "Sprite: A Simple, Cheat-proof, Credit-based System for Mobile Ad-hoc Networks". *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, California, USA, pp. 1987-1997, March 2003.

[7] S. Buchegger, and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol". *Proceedings of 3rd ACM International Symposium on Mobile Ad hoc Networking and Computing (MOBIHOC 2002))*, Lausanne, Switzerland, pp. 226-236, June 2002.

[8] D. B. Johnson, D. A. Maltz and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad hoc Networks" in *Ad hoc Networking*, C. E.Perkins, Ed.: Addison-Wesley Longman Publishing Co., Inc., USA, 2001.

[9] V. Balakrishnan, V. Varadharajan and U. K. Tupakula, "Fellowship: Defense against Flooding and Packet Drop Attacks in MANET". *Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, Vancouver, Canada, pp. 1-4, April 2006.

[10]V. Balakrishnan and V. Varadharajan, "Fellowship in Mobile Ad hoc Networks". *Proceedings of IEEE Security & Privacy in Emerging Areas (SecureCom2005)*, Athens, Greece, pp.225-227, September 2005.

[11] Ping Yi, Zhoulin Dai, Yiping Zhong, Shiyong Zhang, "Resisting Flooding Attacks in Ad hoc Networks". *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing (ITCC 2005),* Las Vegas, USA, pp. 657-662, April 2005.

The 2nd International Conference on Wireless
Broadband and Ultra Wideband Communications (AusWireless 2007)
0-7695-2842-2/07 $25.00 © 2007 IEEE