

Analyzing and Preventing MAC-Layer Denial of Service Attacks for Stock 802.11 Systems

Yihong Zhou
Dept. of Electrical
and Computer Engineering
The University of Texas at Austin
Austin, TX 78712
Email: zyihong@ece.utexas.edu

Dapeng Wu
Dept. of Electrical
and Computer Engineering
University of Florida
Gainesville, FL 32611-6130
Email: wu@ece.ufl.edu

Scott M. Nettles
Dept. of Electrical
and Computer Engineering
The University of Texas at Austin
Austin, TX 78712
Email: nettles@ece.utexas.edu

Abstract—Network survivability is fundamental to information security. Adversaries could compromise network functionality by attacking the physical layer, the medium access control (MAC) layer, or the network layer. Even though security mechanisms for the network layer have been extensively discussed, MAC layer security has not been deeply explored. Leveraging security flaws in the IEEE 802.11 MAC protocol and the salient features of a Mobile Ad Hoc Network (MANET), even an average person with limited knowledge of wireless networks can launch MAC-layer denial of service (DOS) attacks. In this paper, we study two types of MAC layer DOS attacks that can be easily employed by an average person: attacks launched from a single adversary by injecting enormous data flows into the network, and attacks launched from two colluding adversaries by sending enormous data flows directly to each other. We propose new counter measures to defend against these two types of DOS attacks. Since this paper focuses on MAC layer DOS attacks that could be easily executed by an average person, and compromising legitimate nodes is not an easy task, we only consider DOS attacks launched directly from adversaries, instead of from compromised nodes.

I. INTRODUCTION

A Mobile Adhoc Network (MANET) is a collection of mobile nodes communicating with each other without infrastructure assistance. MANETs are used in circumstances such as battlefields, or emergency and rescue missions, in which deploying a fixed network infrastructure is difficult. Because military operations are still the main application of MANETs, network security is especially critical [1], [2]. Generally, there are two concerns in network security: information security (including confidentiality, integrity and non-repudiation) [3], and network survivability (or availability). Without a doubt, network availability is fundamental for information security. Therefore, defending MANETs against malicious attacks is extremely important.

A MANET could be attacked from the physical, media access control (MAC), or network layer [6]. Even though various MANET network layer security mechanisms have been proposed [4], [5], little research has been done on MAC layer security. Moreover, the salient features of MANETs and security flaws in the 802.11 MAC protocol exacerbate MAC layer vulnerability in the sense that even an average person with limited knowledge about wireless networks can launch MAC layer DOS attacks. These attacks can be launched in

two typical approaches:

- First, leveraging unauthorized data transmission in 802.11, a single adversary can intrude into a network, send enormous flows to legitimate nodes, and hence drain the energy of legitimate nodes as well as substantially reduce the available channel capacity for legitimate communications. We call this type of attack a *single adversary attack* (SAA);
- Second, leveraging unfairness present in 802.11, two colluding adversaries may send enormous data flows directly to each other, and hence deplete the channel capacity in their vicinity. We call this type of attack a *colluding adversaries attack* (CAA).

Both attacks pose great threats to MANET survivability, because they can be launched by an average person with standard hardware. To counter SAA attacks, we propose a packet-by-packet authentication scheme so that legitimate nodes can reject data transmission requests from unauthenticated adversaries. Furthermore, to mitigate CAA attacks, we propose several methods such as a fair MAC protocol and using protecting traffic flows.

To our knowledge, little research has been done in regard to MANET MAC-layer DOS attacks, especially those can be employed by an average person. Gupta [6] simulated and analyzed several SAA scenarios. They proposed a fair MAC protocol to mitigate attacks but at the cost of overall throughput degradation. To reduce the throughput degradation, this paper proposes a packet-by-packet authentication scheme.

Since this paper focuses on MAC layer DOS attacks that could be easily executed by an average person, and compromising legitimate nodes is not a easy task, we only consider the attacks launched directly from adversaries, instead of from compromised nodes.

The rest of the paper is organized as follows. In Section II, we overview the vulnerabilities of the IEEE 802.11 MAC protocol. In Sections III and IV, we study two types of attacks, i.e., SAA and CAA, respectively, and propose counter measures. Section V concludes the paper and points out directions for future work.

II. VULNERABILITIES OF THE 802.11 MAC PROTOCOL

IEEE 802.11 [7] is a standard MAC protocol widely used for MANETs. Unfortunately, two security flaws render 802.11-based MANETs vulnerable even to an average person: unauthorized data transmission, and the unfairness problem. In this section, we first briefly introduce the 802.11 protocol and its physical layer model, then we reveal our insights into the two security flaws.

A. The IEEE 802.11 MAC Protocol

The 802.11 Distributed Coordination Function (DCF) standard [7] combines Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with a Request to Send/Clear to Send (RTS/CTS) handshake to avoid collisions. It works as follows.

A node can only transmit when no carrier of a transmitting node is sensed in the vicinity. Otherwise, it has to defer its own transmission until the channel is determined to be idle. The node then requests the channel by sending an RTS to the receiver which, in turn, replies with a CTS. The nodes in the vicinity overhearing the RTS or CTS defer their own transmission for a period that is long enough for the subsequent DATA/ACK exchange. When the RTS/CTS handshake is completed, the sender commences data transmission. The receiver acknowledges the data with an ACK. If no CTS or ACK is received, the sender exponentially backs off, and retransmits the RTS or DATA.

B. Three Physical Layer Ranges

We consider three ranges in the 802.11 physical layer model: the receiving range, R_{rx} , the carrier sense range, R_{cs} , and the interference range, R_{if} . Because they closely relate to the 802.11 unfairness problem, we briefly introduce them in this section.

When a sender transmits, the nodes that are d_{rx} distance away can decode the signal correctly. Hence, the region centering around sender with radius d_{rx} is the receiving range R_{rx} . The nodes that are d_{cs} away from the sender can sense the signal but are not able to correctly decode it, so that the region centering around sender with radius d_{cs} and excluding R_{rx} is the carrier sense range R_{cs} . Finally, the node that is d_{if} away from receiver is able to corrupt the receiving signal by transmitting at the same time. Hence, the region centering around receiver with radius d_{if} is the interference range R_{if} . Xu [9] calculates d_{if} as:

$$d_{if} = \sqrt[4]{SNR} \times d \quad (1)$$

where SNR is the signal to noise ratio of the capture effect, and d is the distance between sender and receiver. If SNR is set to 10dB, d_{if} is 1.78d.

C. Unauthorized Data Transmission

The 802.11 standard does specify two authentication schemes: Open System and Shared Key. But both schemes have security flaws and cannot effectively prevent SAA attacks.



Fig. 1. Network topology for two flows with same traffic direction

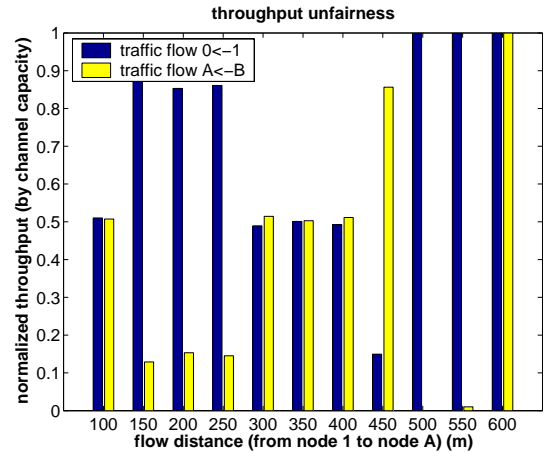


Fig. 2. Unfair throughput of two flows with same traffic direction

Open System, which is the default authentication system, is essentially a null authentication algorithm. Any node requests authentication may become authenticated if authentication-Type at the recipient node is set to Open System Authentication. Even though the other scheme, Shared Key, establishes a mutual authentication relationship between two nodes following a successful authentication exchange, it cannot prevent an adversary from spoofing a legal node's identity, and hence intruding into the network. Moreover, Shared Key is intended for the 802.11 Point Coordination Function (PCF) mode. The 802.11 standard does not specify how Shared Key should be used in the Distributed Coordination Function (DCF) mode, e.g. MANET.

In MANET, all the participating nodes should receive or forward packets unconditionally. Therefore, it gives adversaries the chance of launching SAA attacks.

D. Unfairness

The other 802.11 security flaw is its unfairness, which leaves MANET open to CAA threats. That is, colluding adversaries can suppress an intended traffic flow by moving to appropriate positions and sending enormous data flows directly to each other.

To gain an understanding of unfairness, we create the network topologies illustrated in Fig. 1, and measure the throughput of each flow with increasing flow distance (d_f) in the ns2 simulation environment [8]. Both traffic flows, Tr_{01} and Tr_{AB} , send data packets at full channel capacity. Fig. 2 is the simulation results for one of the four traffic direction combinations: node 1 sends to node 0, and node B sends to node A. The other three combinations demonstrate the same unfairness effects. We use this one to gain some insight into the problem.

Fig. 2 shows that the unfairness problem occurs at three different flow distances: when d_f is 150m to 250m, Tr_{01}

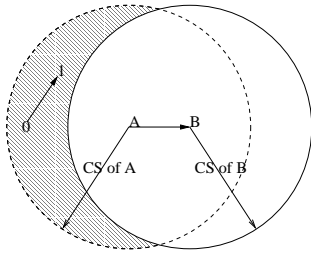


Fig. 3. Unfairness caused by the difs/eifs timing scheme

suppress Tr_{AB} ; when d_f is 400m, Tr_{AB} suppress Tr_{01} ; when d_f is 500m to 550m, Tr_{AB} is suppressed again by Tr_{01} . This unfairness is caused by two basic mechanisms in the 802.11 MAC protocol: the difs/eifs timing scheme, and the exponential backoff mechanism. They are explained in the following sections.

1) *difs/eifs Timing Scheme*: To realize different channel access priorities, 802.11 uses three interframe spaces: the short interframe space (sifs), the DCF interframe space (difs), and the extended interframe space (eifs). They are defined by the following equations:

$$difs = sifs + 2 \times SlotTime \quad (2)$$

$$eifs = sifs + \frac{ACKframesize}{BasicRate} + difs \quad (3)$$

Sifs is the time delay introduced by the physical layer hardware. Difs and eifs are the period of duration for a node to determine the channel status. Obviously, eifs is much longer than difs, and difs is longer than sifs. Therefore, a node that waits for a period of sifs duration to transmit the next frame has the highest channel access priority, while a node that waits for a period of eifs duration to transmit the next frame has the lowest channel access priority.

The 802.11 specifies that a node needs three steps to commence the next data packet transmission: wait for a period of difs/eifs duration to determine the channel status to be idle, resume backoff, and start RTS/CTS handshake. Furthermore, if a node correctly decodes the last frame detected on the medium, it needs a period of difs duration to determine the channel status to be idle. However, if a node cannot correctly decode the last frame detected on the medium, it needs a period of eifs duration to determine the channel status to be idle. In another words, if a node can correctly decode the last detected frame, it has higher channel access priority than the one who can not correctly decode the last detected frame. As a consequence, unfairness is resulted. Fig. 3 and 4 illustrate the scenario.

The scenario illustrated in Fig. 3 corresponds to the flow distance 450m in Fig. 2. Node 0 and node 1 are in R_{cs} of A but out of R_{cs} of B. When A is transmitting, node 0 can detect but can not correctly decode the signal. Since node 0 is out of R_{cs} of B, it cannot ever detect the signal from B.

Assume node A completes a data transmission at t_0 . Node 0 resumes backoff at t_1 ($t_0 + \Delta_{eifs}$), because it needs a period

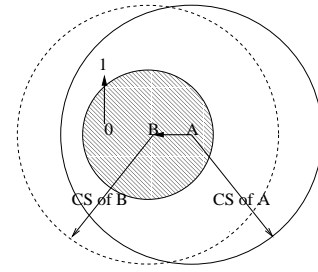


Fig. 4. Unfairness caused by the difs/eifs timing scheme

of eifs duration to determine channel status. Node A resumes backoff at t_2 ($t_0 + \Delta_{ACK} + \Delta_{difs}$) because it needs Δ_{ACK} to receive ACK from B and a period of difs duration to determine channel status. According to equation 3, t_1 and t_2 are almost the same instance. Hence both node 0 and node A have equal opportunity to access the channel in this case.

Now, consider the case when node 0 completes a data packet transmission at instance t_0 . It would receive an ACK at t_1 ($t_0 + \Delta_{ACK}$), and resume backoff at t_2 ($t_1 + \Delta_{difs}$). Since node A can detect but cannot correctly decode the ACK from node 1, it resumes backoff at t_3 ($t_1 + \Delta_{eifs}$). Because time t_3 is much later than t_2 , node A has much less chances of winning the channel, which leads to severe unfairness.

Fig. 4 depicts another unfairness scenario caused by the difs/eifs timing scheme, which corresponds to the flow distances of 150m, 200m, and 250m in Fig. 2. In this case, node 0 is in R_{rx} of B so that it can correctly receive an ACK signal from B. Node 1 is in R_{cs} of A, so that node A can detect but not correctly decode ACKs from node 1. Assume A completes a data transmission and B replies with an ACK, node 0 and A have the same chance of accessing the channel because both of them can correctly receive the ACK signal from B.

However, unfairness happens when node 0 and 1 are transmitting. Assume node 0 completes a data transmission at t_0 . It starts backoff after receiving the ACK from node 1 and waits for a difs duration, which is at instance t_1 ($t_0 + \Delta_{ACK} + \Delta_{difs}$). Since node A can detect the ACK signal from node 1 but can not correctly decode it, node A starts backoff at t_2 ($t_0 + \Delta_{ACK} + \Delta_{eifs}$). As a consequence, node A always losses the channel access competition.

2) *Exponential Backoff*: The exponential backoff mechanism, used for resolving collisions in the 802.11 MAC protocol, is another source of the unfairness problem. Exponential backoff occurs when the sender does not receive a CTS or an ACK from the receiver. Therefore, the sender retransmits its RTS or data packet with an exponentially increasing backoff window, which leads to lower and lower chances of accessing the channel. Fig. 5 illustrates this scenario, which corresponds to flow distances 500m and 550m in Fig. 2.

In Fig. 5, both node 0 and 1 are in R_{cs} of B but out of R_{cs} of A, so that node B can detect the signal from node 0 and 1 but A cannot.

When node 0 and 1 are transmitting, A may send an RTS to B since it cannot detect the signal, and hence believes the

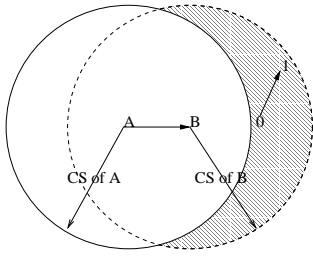


Fig. 5. Unfairness caused by the exponential backoff mechanism

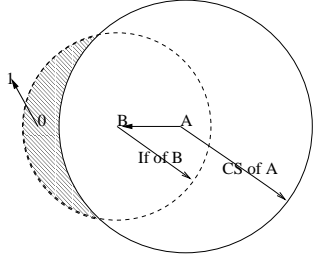


Fig. 6. Unfairness caused by interference

channel to be idle. But node B can not reply with a CTS since it detects the carrier, and determines the channel to be busy. A has to exponentially increase the backoff window and wait for a longer time to retransmit the RTS. As long as the traffic load of Tr_{01} is heavy enough, A would have very limited chances of receiving a CTS from B. As a result, the throughput between node A and B is almost zero.

The unfairness caused by exponential backoff is even worse than that caused by the difs/eifs timing scheme, as shown in Fig. 2. When the flow distance is 500m, Tr_{AB} should have a higher throughput than Tr_{01} because of the difs/eifs timing scheme. But on the contrary, Tr_{01} suppresses Tr_{AB} because of the exponential backoff effect. Exponential backoff always results in the suppressed flow throughput being zero, while the difs/eifs timing scheme does not.

3) *Interference*: Interference is the third reason that may cause severe unfairness, which is not reflected by our above experiment.

According to equation 1, R_{if} expands with the increasing of the distance between sender and receiver d . Therefore, at a certain threshold of d , both R_{cs} of the sender and R_{rx} of the receiver may no longer cover the whole area of R_{if} . The hidden nodes, that reside in R_{if} but out of R_{cs} , may commence simultaneous transmission, which would generate interference strong enough to corrupt the receiving signal at the receiver. Therefore, the receiver would not reply with either a CTS or an ACK. As a consequence, the sender has to exponentially backoff and retransmit either the RTS or the data packet. This scenario is illustrated by Fig. 6.

In Fig. 6, node 0 is in R_{if} of the receiver B but out of R_{cs} of the sender A. When A is sending packets to B, node 0 may commence transmission which results in either the RTS or data signal being corrupted at B. A has to exponentially backoff and retransmit because it does not receive a CTS or ACK from B.

As long as the traffic load of Tr_{01} is heavy enough, B would have no chance of correctly receiving a packet, which leads Tr_{AB} to almost zero throughput.

III. THE SINGLE ADVERSARY ATTACK

Based on the previous analysis, we introduce two kinds of MAC-layer DOS attacks: the *Single Adversary Attack* (SAA) and the *Colluding Adversary Attack* (CAA). In this section, we focus on SAA attacks, and propose a counter measure of packet-by-packet authentication. CAA is discussed in the next section.

A. Attacking Methods

To launch a SAA, a single adversary simply intrudes into the network, and sends enormous data flows to any legitimate node. Therefore, channel capacity and the participating nodes' energy would be depleted. SAA can generate global impact on the network. For example, if the intended destination is multiple hops away, the nodes along and in the vicinity of the routing chain would all be affected.

In MANETs, the open medium and unauthorized data transmission make SAA so easy to be launched that even normal person can execute it without any restrictions. It leaves MANET with a great security threat.

B. Packet-By-Packet Authentication

To counter SAA attacks, we propose a packet-by-packet authentication scheme. We base our authentication scheme on a key shared among the legitimate nodes. The shared key is presumed to have been delivered to the legitimate nodes through a secure channel.

The authentication process is integrated into each RTS/CTS exchange. So that a node rejects receiving and forwarding a data packet by not replying with a CTS if the received RTS has the wrong authentication information.

To effectively counter replay and spoofing attacks, we introduce a timestamp. For each sender/receiver pair, the timestamps in the authentication messages are nondecreasing. If the time resolution is not high enough to distinguish each RTS packet, a sequence number can be added. The sequence numbers are always increasing. Particularly, the sequence number for the first RTS between a sender/receiver pair is set to be zero. The details of the authentication scheme are discussed below.

1) *Authentication Scheme*: First, each node maintains two tables. One table records the latest RTS information received from each sender. The other table records the latest RTS information sent to each receiver. Both tables use the same format, which is shown in Table III-B.1. Node ID is the sender's (or the receiver's) MAC address. TS and SN are the timestamp and the sequence number of the most recent RTS packet respectively.

When a sender A is about to send an RTS packet, it timestamps the RTS, increments the sequence number by 1, and encrypts the source MAC address sID , destination MAC address dID , TS, SN with the shared key k . The encrypted message is attached at the end of the RTS packet:

Node ID	Time Stamp	Sequence Number
Node B	TS_B	SN_B
Node C	TS_C	SN_C
...

TABLE I
TIMESTAMP TABLE

$$A \rightarrow B : \{RTS, E\{sID, dID, TS, SN\}_k\}$$

Upon receiving the RTS, the receiver B decrypts the cypher with the shared key, verifies sID and dID, checks the freshness of TS, and compares SN with the old one stored in the table. The RTS is accepted if TS is not less than and SN is greater than the previous ones. If B receives an RTS with a zero sequence number, it assumes that this is the first RTS from A or that A has been reset for some reason, so that only timestamp is checked in this case. Finally, B replies with a CTS packet with an encrypted attachment. The TS and SN are the same as those in the received RTS encryption attachment:

$$B \rightarrow A : \{CTS, E\{sID, dID, TS, SN + 1\}_k\}$$

The sender decrypts the encrypted attachment, verifies sID, dID, TS as well as SN+1. The sender can commence data transmission if both sender and receiver are authenticated.

2) *Security Analysis*: Our security scheme comes at the cost of transmission efficiency and energy consumption.

Because of the additional encryption/decryption messages, transmission overhead increases. Assume TS and SN are 4 bytes each, the total encrypted attachment is 20 bytes (6 bytes each for sID and dID). Therefore, the overhead increases about 40 bytes (RTS and CTS) per data packet transmission.

Energy consumption is another tradeoff. Extra energy is consumed in two ways: the network interface transmit/receive of 20 bytes of authentication overheads [10], and the 20 bytes of message encryption/decryption computation [11], [12].

The limitation of the proposed scheme is that it can not authenticate broadcast packets since no RTS/CTS handshaking can be leveraged.

Key management is critical for the proposed authentication scheme. The shared key is presumed to have been delivered to the participating nodes via a secure channel. To counter brute force attacks, the key length should be longer than 64 bits so that it cannot be compromised in a short period of time [13].

3) *Enhancing Key Management*: Obviously, the biggest drawback of the above scheme is the shared key. If one node is compromised, the whole system is invalidated. To enhance security, we propose the asymmetric (public/private) key management scheme, in which each node maintains a private key that is kept secret, and a public key that could be known by all the other nodes and can be proved by the certification issued by a CA (Certificate Authority). Since it is difficult to establish a centralized CA in MANETs, a distributed CA can

be established using threshold cryptography technologies [14]. Moreover, if public key revocation and update are rare, the certification could be issued before the nodes join the network so that they can leverage a CA outside of the MANET. We assume the CA for a MANET only issues certification to those who are permitted to join the network, e.g legitimate nodes.

The key management scheme works as follows. First, each sender/receiver pair negotiates a temporary session key k . All the subsequent data transmissions would base on k until it expires. The negotiation process is described below:

$$A \rightarrow B : \{sID, dID, E\{sID, dID, TS\}_{P_A^{-1}}, \\ Certification\}$$

$$B \rightarrow A : \{sID, dID, E\{E\{sID, dID, TS, k, T\}_{P_B^{-1}}\}_{P_A}, \\ Certification\}$$

where P_x and P_x^{-1} are the public key and private key of entity x respectively. The requester A sends B negotiation request which contains its certification as well as the digital signature of sID, dID and TS. B replies a random generated session key k , which is signed by it's private key and encrypted by A's public key. A decrypts the reply message with its private key P_A^{-1} , verifies the signature, retrieves k and its life time T . Thus a session key k is distributed to both parties of the transmission, and the key is known only to the two parties.

IV. THE COLLUDING ADVERSARIES ATTACK

CAA is another approach to attack MANETs by a normal person. Leveraging the 802.11 unfairness problems, colluding adversaries can suppress the intended traffic flow by sending enormous data flows directly to each other. In this section, we first analyze different CAA attack scenarios, then propose several methods to mitigate the attacks.

A. Attacking Methods

To introduce CAA attacks, we first define the concept of the Attacking Region (AR). The AR is always associated with a particular traffic flow Tr_x . Once another traffic flow Tr_y moves into the AR of Tr_x and commences transmission with enormous amount of traffic, the throughput of Tr_x would be suppressed. According to the analysis in section II, four ARs exist around a particular traffic flow Tr_x :

- AR_{sender} : is caused by the 802.11 difs/eifs timing scheme, and resides at the sender's side. Colluding adversaries in this area have higher priority of accessing the channel than Tr_x 's sender. AR_{sender} is defined as sender R_{cs} minus receiver R_{cs} , which is illustrated as the shadow part in Fig. 3.
- $AR_{receiver}$: is also caused by the difs/eifs timing scheme, but resides at the receiver's side. It is the area of receiver R_{rx} , which is the shadow part in Fig. 4.
- $AR_{exppbackoff}$: is generated by the 802.11 exponential backoff mechanism, as explained in section II. It is the area of receiver R_{cs} subtracting sender R_{cs} , which is illustrated as the shadow part in Fig. 5.

- $AR_{interference}$: the area of receiver R_{if} minus sender R_{cs} , as shown in Fig. 6. AR_{if} is caused by interference around receiver and exists only when the distance between sender and receiver is greater than a threshold.

As long as colluding adversaries move to one of the ARs of the targeted traffic flow, they can launch effective DOS attacks. In the following sections, we analyze different attacking methods: attack at the sender side, attack at the receiver side, attack of multihop traffic flows, and attack of multiple traffic flows. To facilitate the description, we assume node A and B are the legitimate sender and receiver respectively. Node 0 and node 1 are the colluding adversaries.

1) *Attack at the Sender Side*: If two colluding adversaries, node 0 and 1, want to launch DOS attacks from the sender's side, they just move to AR_{sender} , and hence form the network topology depicted by Fig. 3. As long as the traffic of Tr_{01} achieves full channel capacity, the legitimate traffic flow Tr_{AB} will be suppressed because of the difs/eifs timing scheme.

If node A and B change their traffic direction, the network topology becomes the one illustrated by Fig. 5. In this case, the colluding adversaries are in $AR_{expbackoff}$ of Tr_{AB} , and they suppress Tr_{AB} by leveraging the exponential backoff mechanism.

2) *Attack at the Receiver Side*: It is easier to launch the DOS attacks from receiver side, since the other three ARs are around the receiver.

To launch exponential backoff attacks, two colluding adversaries should both move to $AR_{expbackoff}$ so that the network topology depicted by Fig. 5 can be constructed. If nodes A and B change traffic directions, the colluding adversaries would be in AR_{sender} . The throughput of the legitimate traffic would also be suppressed as illustrated by Fig. 3.

Alternatively, to leverage the difs/eifs timing scheme, the colluding sender should move to $AR_{receiver}$ and the colluding receiver can be anywhere else except $AR_{receiver}$ and R_{rx} of the legitimate sender. Therefore, the network topology depicted by Fig. 4 is formed.

Once the distance between sender and receiver is beyond the interference attacking threshold, interference attacks can be launched. If the colluding sender moves to $AR_{interference}$, the network topology depicted by Fig. 6 can be formed. It would better if the colluding receiver is in $AR_{expbackoff}$ of the legitimate receiver so that Tr_{AB} would also be suppressed even if they change traffic directions, which is illustrated by Fig. 3.

3) *Attacking Multihop Traffic Flows*: Attacking multihop traffic flows results in worse effects than attacking one hop traffic. Two colluding adversaries can attack any hop of the chain as long as they move to the appropriate ARs. To investigate the multihop traffic throughput under CAA attacks, we create the network topologies as Fig. 7.

Node A, B and C form a two-hop traffic flow Tr_{ABC} with hop distance 130m. Their coordinates are 600, 730 and 860 respectively. Node 0 and 1 are colluding adversaries separated by 50m. We gradually move Tr_{01} from position 0 to 1450,

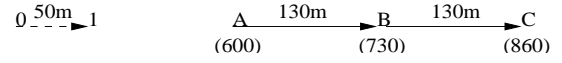


Fig. 7. network topology of attacking multihop traffic flow

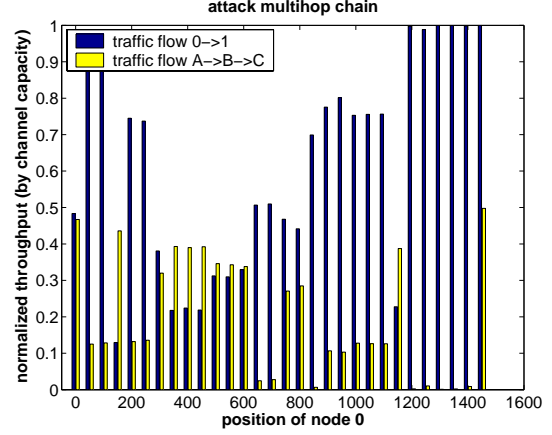


Fig. 8. attacking multi-hop traffic flow

and measure the throughput of the two flows. The simulation results are shown in Fig. 8.

Even though it is possible for the colluding adversaries to attack each individual hop, the best position to launch the attacks is at the destination (node C) side. Starting from position 1200, the adversaries cause the receiver B (or C) to be unable to reply with a CTS, and the sender A (or B) has to exponentially backoff. Therefore, no data packet could be sent to the final destination C. This situation continues until node C moves out of the R_{cs} node 0, where node 0 reaches position 1450.

Leveraging the exponential backoff mechanism, the colluding adversaries cause data packet dropping at the intermediate nodes. Hence, not only is the traffic suppressed, but also the energy of forwarding the packets is wasted. This is the worst effects that can be result from CAA attacks.

4) *Attacking Multiple Traffic Flows*: It is difficult to attack multiple traffic flows at the same time because the attacking regions of each traffic may have no intersection. Even if intersection exists, severe unfairness may not be achieved. Fig. 9 illustrates one of the scenarios:

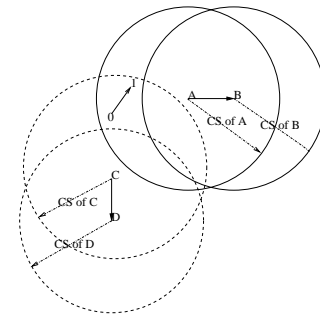


Fig. 9. attacking multiple traffic flows at the same time

Two colluding adversaries node 0 and 1 want to attack Tr_{AB} and Tr_{CD} at the same time. They move to the area which is at the intersection of the two AR_{sender} , and commence data transmission with full channel capacity. The simulation results show that Tr_{01} cannot suppress either Tr_{AB} or Tr_{CD} . On the contrary, the throughput of Tr_{01} is almost zero.

In this scenario, node 0 and 1 are in R_{cs} of both A and C, so that node 0 has to compete the channel with both of them. Since node A and C are far away enough, they cannot detect the carrier of each other. Hence, they may transmit data packet simultaneously without interfering with each other. As a consequence, node 0 is deferred either by A or C's transmission. The chances of channel access for node 0 is greatly reduced.

B. Counter Methods

There are several ways to mitigate CAA attacks. Some of them depend on attack detection mechanisms.

1) *Fair MAC Protocol*: The unfairness problem is the basic reason that leaves MANETs open to CAA attacks. Therefore, adopting a fair MAC protocol would be an efficient approach to mitigate CAA attacks. Even though, the network throughput would degrade to some extent, the legitimate traffic flow will not be entirely suppressed.

2) *Protecting Traffic Flow*: From the previous analysis, we observe that it is hard to attack multiple traffic flows at the same time. That is, there should be no other traffic flows exist around adversaries traffic flow. In another word, CAA attacks should be launched at the edge of the network. Therefore, once CAA attacks are detected, another protecting traffic flow can be triggered in the vicinity of the adversaries so that the network topology illustrated by Fig. 9 can be formed. Once the conditions for CAA attacks are broken, the legitimate traffic flow is protected.

3) *Distance Adjustment*: The larger the distance between sender and receiver, the wider area the ARs are. When the distance is beyond a certain threshold, interference attacks could be launched. In this case, the CAA attacks can be avoided by moving sender and receiver close to each other.

V. CONCLUSIONS

In this paper, we studied two typical MAC layer DOS attacks, namely, SAA and CAA, which could be easily employed by an average person with standard hardware.

For SAA, a single adversary sends enormous data flows to legitimate nodes, and hence drains the energy of the legitimate nodes and substantially reduces the available channel capacity available to legitimate nodes. To counter SAA attacks, we proposed a packet-by-packet authentication scheme.

For CAA, two colluding adversaries suppress the legitimate traffic by sending enormous flows directly to each other. To mitigate CAA attacks, we proposed several approaches, such as using fair MAC protocols, introducing protecting traffic flow, and adjusting the distance.

Our contribution is that we identified the potential threats to MANET security. Especially, threats that may come from an average person with limited knowledge about wireless networks and standard wireless equipment such as laptops, PDAs, etc. More sophisticated attacks such as DOS attacks launched from compromised nodes are a possible future research direction.

REFERENCES

- [1] Frank Stajano, Ross Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", *AT&T Software Symposium 1999*.
- [2] Nikita Borisov, Lan Goldberg, David Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11", *Proceedings of Mobicom 2001*. Rome, Italy, 2001.
- [3] Lidong Zhou, Zygmunt J. Haas, "Securing Ad Hoc Networks", *IEEE Network*, vol.13, issue 6, Nov.-Dec. 1999, pp 24-30.
- [4] Panagiotis Papadimitratos, Zygmunt J. Haas, "Secure Data Transmission in Mobile Ad Hoc Networks" *Proceedings of the 2003 ACM workshop on Wireless security*, Sept. 2003.
- [5] Yih-Chun Hu, Adrian Perrig, David B. Johnson, "Security: Ariadne: a secure on-demand routing protocol for ad hoc networks", *Proceedings of the 8th annual international conference on Mobile computing and networking*, Sept. 2002.
- [6] Vikram Gupta, "Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks", *Proceedings of MILCOM 2002*, vol. 2, 7-10 Oct. 2002, pp 1118-1123.
- [7] Wireless LAN Medium Access Control and Physical Layer Specifications, Aug. 1999. IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee).
- [8] CMU Monarch Extensions to ns <http://www.monarch.cs.cmu.edu/>
- [9] Kaixin Xu, M. Gerla, Sang Bae, "How Effective is the IEEE 802.11 RTS/CTS Handshke in Ad Hoc Networks?" *GLOBECOM '02. IEEE*, vol 1, 17-21 Nov. 2002 pp. 72 - 76
- [10] Laura Marie Feeney, Martin Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment", *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol 3, 22-26 April 2001
- [11] T. Burd and R. Brodersen, Processor Design for Portable Systems, *Journal of VLSI Signal Processing*, vol. 13, no. 2, pp. 203222, August, 1996
- [12] Trevor Pering, Tom Burd, Robert Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms", *Proceedings of International Symposium on Low Power Electronics and Design, 1998* Aug. 1998 Pages:76 - 81
- [13] Matt Curtin, Justin Dolske, "A Brute Force Search of DES Keyspace", <http://www.interhack.net/pubs/des-key-crack/>.
- [14] D. Dhillon, T.S. Randhawa, M. Wang, L. Lamont, "Implementing a Fully Distributed Certificate Authority in OLSR MANET", *WCNC 2004*