DRETA: Distributed Routing Evidence Tracing and Authentication intrusion detection model for MANET

Insha Majeed
Dept. of IT, National Institute of Technology,Srinagar, J&K, India
insha333@gmail.com

Insha Altaf
Dept. of IT, National Institute of Technology, Srinagar, J&K, India
insha.altaf39@gmail.com

***Abstract--*To give a productive and low-overhead assurance to keep forwarded routing messages from being fashioned, we propose the utilization of the Distributed Routing Message Tracing and Authentication intrusion detection model (DRETA). In the first place, DRETA has a distributed design, in which every hub has a detector to screen and accept routing messages with one another autonomously. Additionally, not at all like numerous safe routing protocols, DRETA has a free layer, isolated from the network layer, which intercepts routing messages and does not require the change of the routing protocol. Second, DRETA utilizes symmetric keys, which require much lower calculation overhead than public keys, to give authentication administrations to all routing messages. DRETA embraces one-way keychain and delay key revelation strategies to permit symmetric keys to be conveyed in public channels like Public Key Infrastructure (PKI) does. Third, DRETA proposes the utilization of Validation Messages (VMs), which utilize HMAC to secure the integrity of forwarded messages. While the sender forwards a forwarded message to the receiver, the sender additionally declares the Previous Forwarder (PF), the individual who already sent the message to the sender. DRETA utilizes Previous Forwarder to engender the keys and VMs to follow and guarantee the exactness of routing proof conveyed in the forwarded messages, utilizing minimal message overhead. In this way, DRETA is an adaptable, effective, and nonspecific arrangement which utilizes low overhead to ensure the integrity and authenticity of sent routing messages in MANET. We actualize DRETA on two agent routing protocols, AODV and OLSR.**

Keywords— Access control, AODV, storage node, Optimized Link State Routing, Topology Control, hop, solar, rq, nodes, Mamet.

I. INTRODUCTION

A. AODV

AODV is the agent request on-demand routing protocol in MANETs. It has little size routing messages, which contain just routing data for the source and destination; a sequence number is utilized to show the freshness of the information [4].

While a source node S requires a course toward a destination node D, node S broadcasts a RREQ message rq to ask for the route. After accepting rq, the receiver tosses rq on the off chance that it beforehand got the same RREQ message; this is refined by verifying whether it had gotten any RREQ with the same source address and the same RREQ ID as rq. Something else, the receiver stores the opposite route towards the source if rq has a higher source sequence number or an equivalent sequence number (with littler hop count toward the source) than that the receiver had [2].

In the event that the receiver upgrades the converse route by rq and does not have a legitimate route to the destination D, it will rebroadcast rq and expand the hop count of rq by one. In the event that the receiver has a substantial route toward the destination D or the collector itself is the destination D, it will produce a RREP message rap and unicast rp along the opposite route toward the source S.

While accepting rp, the receiver stores the forward route toward the destination D if rp has a higher destination sequence number than the receiver had or an equivalent sequence number with littler hop count toward the destination. In the event that this is thus, the receiver additionally unicasts rp along the opposite course back to the source S and expansions hop count of rp by one (unless rp comes to the source S)[3].

After a course has been set up in the middle of S and D, if one transitional hub R1 sees that it can't achieve another node R2 (which was beforehand reachable from R3), then R3 will show a RERR to report this broken connection. Furthermore, in order to maintain local connectivity, each node will send a Hello message every second to announce its existence if it does not send any RREQ, RREP, and RERR within second.

B. OLSR

OLSR is a representative proactive link state routing protocol of MANET. It uses two periodical routing messages, Hello and Topology Control (TC) messages, to establish and maintain the routing topology. OLSR proposes Multi-Point Relay (MPR) to minimize message flooding. A set of MPR nodes is a subset of 1-hop neighbors that can connect all 2-hop neighbors [5].

In OLSR, nodes first exchange Hello messages and establish their 1-hop neighbor lists. Subsequently, nodes compute their 2-hop neighbor lists from their 1-hop neighbor lists. Furthermore, nodes compute their MPR sets from their 1-hop and 2-hop neighbor lists, and announce their MPR sets in their Hello messages.

C. ONE-WAY KEY CHAIN

A one-way key chain[16] is produced by repeatedly applying a one-way hash function H on a random number r for n times, and the last computed number is the initial key: r = Kn, H(r) = Kn-1, ... , K0 = H(K1). It is computationally infeasible to get x from H(x). Given Ki, anyone can compute Ki-1 to K0 from Ki, but cannot compute Ki+1 to Kn from hash function H.

Keys in a one-way key chain are symmetric keys, so they can be utilized to a produce keyed Hash Message Authentication Code (HMAC) [4]. HMAC is the signed hash value of the message and is utilized to secure the integrity of the message. Since HMAC utilizes symmetric keys, HMAC requires much lower computational overhead than public key based digital signatures. Secure protocols[15]using digital signatures require 100 times the computation for signature generation and 10 times the computation for signature verification than HMAC does. Because of the limited computational resources for a node in MANETs, using HMAC can prevent denial of service attacks caused by the high computation overhead of digital signatures [6].

D. DELAY KEY DISCLOSURE

Delay key divulgence [2] is an ideal key distribution administration intended for symmetric key based HMAC, particularly for a completely distributed environment, for example, a MANET. In the wake of processing a restricted keychain, a node N discharges its Koi to start a boot strapping procedure. Given a period interim I, hub N discharges its Ki-1 and utilizations Ki to create a HMAC, affixed with the protected broadcast message M. The current key Ki is utilized to sign the message M (like a private key in PKI), and the last lapsed key Ki-1 is utilized to approve the current key Ki (like a public key in PKI). In the following time interval i+1, node N discharges the lapsed Ki to permit message receivers to authenticate M by MAC and Ki. Since Ki can deliver Ki-1 from a hash function, Ki-1 can be utilized to validate Ki. After some time, the current key (like a private key) and last terminated key (like a public key) continue redesigning taking after the keychain[7].

II. ATTACK MODEL OF FORWARDED ROUTING MESSAGE

MANET depends intensely on the distributed routing service, which accept that nodes are helpful and fair. An attacker can without much of a stretch disturb the routing service in MANET in two ways: (1) drop forwarded packets or (2) Send routing messages containing forged content [10].

Drop forwarded packets

A narrow minded node can purposefully drop all forwarded packets experiencing it (Black Hole). On the other hand, it can specifically drop those packets from or toward certain nodes it hates.

Forge routing messages

This class of attacks can effectively influence the routing service of different nodes, especially when the attacks degenerate casualties' routing tables. The fundamental attack method of this classification is mimic, and along these lines authentication is the key requirement for secure routing components. A replay attack is a variety of mimic and might go through authentication protection.

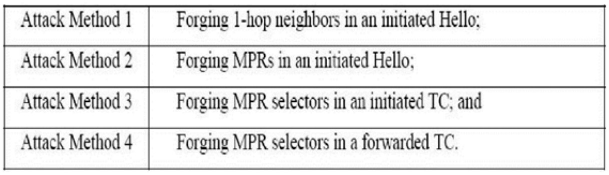
A. AODV VULNERABILITIES

AODV relies on RREQ and RREP to establish routes and routing tables. Sequence number, hop count, source, and destination addresses are critical fields in RREQ and RREP. If a source or destination node generates an incorrect RREQ or RREP (C type), then the incorrect message will become invalid [11].

However, if an intermediate node modifies or creates a RREQ or RREP (A or B type) containing carefully manipulated forged information, such as lower hop counter or higher sequence number, the forged message can change the receivers' routing tables to create virtual links or annul existing links. Since an attacker can utilize several forged messages to manipulate existing and un-existing routes, the attacker can launch more sophisticated man-in-the-middle or denial-of-service attacks, or paralyze the entire network [8].

B. OLSR VULNERABILITIES

In OLSR, the computation of routing tables depends on these fields: 1-hop neighbor lists and MPRs in Hello messages and MPR selectors in TC messages. If a node S originates an incorrect Hello and TC message (C type) containing forged information (such as adding a node A to or deleting a node A from the 1-hop neighbor list, MPR list, or MPR selector list), node S can use the first three attack methods in Fig. 1 to corrupt other nodes' routing tables and cause severe damage of the network topology. Because the forged content is similar to the changes in routing topology caused by mobility, it is challenging to detect these kinds of attacks.



Attack Method 1	Forging 1-hop neighbors in an initiated Hello;
Attack Method 2	Forging MPRs in an initiated Hello;
Attack Method 3	Forging MPR selectors in an initiated TC; and
Attack Method 4	Forging MPR selectors in a forwarded TC.

Figure 1: Attack methods in OLSR

If an intermediate node M modifies or creates an OLSR routing message (type A or B) containing forged information, it will also cause severe routing damage. Hello messages are easier to authenticate because they are broadcast only by the originator with TTL=1. However, TC messages, which are flooded to the entire network, are forwarded by many nodes, and thus, the forwarded TC message's sender can be any other nodes. As a result, node M can attack any other node in the network by forging their TC messages (method 4 in Fig. 1) and this corrupted information will be propagated to the entire network since TC messages propagate and cause wider ranges of damage than those using the other three attack methods in Fig.1 [9].

III. DESIGN CHALLENGES

A. CHALLENGES OF PROTECTING FORWARDED ROUTING MESSAGE

Numerous routing protocols in MANETs broadcast flooding messages to set up and keep up routing administrations. In the event that a noxious middle of the road hub alters the substance of forwarded messages, the vindictive node can control the adjusted messages to change other nodes' routing tables and dispatch serious routing attacks. To keep this sort of new routing attack, the security instrument requires following the exactness of the substance of forwarded messages. Be that as it may, following the substance of forwarded messages is a testing and exorbitant errand. In the event that a progressive wanton checking methodology is connected, high calculation and message overhead happen, and this would not be moderate for MANETs because of constrained calculation and bandwidth assets. Furthermore, since MANET is a completely dispersed environment, following flooding messages might likewise bring about versatility issues. Subsequently, another appropriated methodology is craved for ensuring the integrity of forwarded routing messages, and the new approach must be versatile, effective, and have low message and computational overhead [12].

B. CHALLENGES OF DEVELOPING RELIABLE AND EFFICIENT IDS

A MANET is an appropriated and profoundly dynamic network environment. Versatility and generally questionable wireless channels result in an unusual dynamic network topology. In light of the completely conveyed network, setting up a brought together node which can gather the greater part of the network traffic is not attainable. Also, mobile nodes have generally restricted power and bandwidth requirements, so they can't convey high overhead security protection.

A perfect intrusion detection model in MANET ought to first have a dependable, distributed, low-overhead, message gathering and trading component. The mechanism ought to additionally adjust to changes in the system topology and endure message misfortune. Second, the model ought to be moderate for low calculation power devices. Third, the model ought to perform ongoing insurances since the routing topology might change rapidly and the assault harm might likewise spread generally rapidly. Last, the model ought not to produce high false positives and negatives as to new routing attacks [13].

IV. DRETA

To resolve these unique challenges in MANETs, we propose the use of the Distributed Routing Message Tracing and Authentication intrusion detection model (DRETA) for ensuring the accuracy of flooded routing messages. DRETA adopts a one-way key chain [15] with delay key disclosure [2] to protect the integrity of flooding routing messages with minimal message and computational overhead. DRETA also authenticates the sender of routing messages and prevents replay attacks. In addition, DRETA adapts to mobility and unpredictable topology changes in MANETs and instantly verifies messages to prevent routing attacks [14].

A. DISTRIBUTED AND INDEPENDENT DETECTION ARCHITECTURE

DRETA proposes a fully distributed detection architecture, which is a scalable approach and allows detectors to monitor routing messages with minimal overhead. DRETA also introduces an independent detection mechanism to support various routing protocols in MANETs without modifying them.

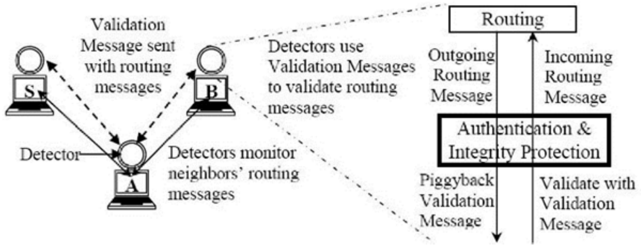

Validation Message sent with routing messages
S
B
Detectors use Validation Messages to validate routing messages
Detector
A
Detectors monitor neighbors' routing messages
Routing
Outgoing Routing Message
Incoming Routing Message
Authentication & Integrity Protection
Piggyback Validation Message
Validate with Validation Message

Figure 2: Distributed detectors use Validation Messages to validate routing messages

B. DISTRIBUTED MESSAGE VALIDATION

DRETA adopts one-way key chaining with delay key disclosure to perform public key style key management without having the high computational overhead of a public key algorithm. By using a sending buffer and Validation Messages (VMs), DRETA performs instant message validation with one-way key chaining and has low computational overhead and message delay.

One-way key chain with Delay Key Disclosure

Most of the secure routing protocols in MANETs use digital signatures, which requires high computational overhead. Symmetric key techniques are fast, but not scalable because each pair of nodes needs to share a shared key. To solve this problem, DRETA adopts one-way key chaining [11] technique with delayed key disclosure.

A node N first generates a series of key chains [K1, K2, and KN], in which Ki can produce Ki-1 but cannot produce Ki+1. Then node N releases K1 as its first public key and uses K2 as its first private key. As K2 expires, N releases K2 as new public key and uses K3 as new private key. Then a neighbor of N, node A, can validate a message which was sent by N and signed by K2. Also, node A can use K1 to validate K2 because K2 can generate K1 by the one-way hash function. So, in a time frame it, node N can use Ki as the private key and Ki-1 as the public key. Thus, with one-way key chain and delay key disclosure, DRETA can perform public key-style key management using symmetric key computation overhead [9].

Distributed Instant Message Validation with Sending Buffer

DRETA proposes a new message validation technique to support instant flooding message validation. By adding a little message validation delay to the first flooding message, this technique uses symmetric keys in the key chain as public and private keys in order to instantly validate forwarded messages.

DRETA proposes a Validation Message (VM) to protect the integrity of routing messages. While a routing message is being sent, DRETA buffers this message, signs it with a VM and sends the VM first. Then the routing message is sent when the key of the VM expires. For example, while a node N has a new routing message M to be sent, N buffers the message M and generates a VM for M with a unique Ki, which is the current N's private key and is used for M only. N first sends M's VM to those prospective receivers and then sends message M with Ki after Ki expires. When the receivers receive M and Ki, they can use M's VM to validate M instantly. Ki is usually attached with the next Validation Message, and Ki also functions as the public key of Ki+1, which is a new private key of a new VM. The valid time of a key should be much longer than the transmission time, but be short enough to represent routing changes [8].

Consider flooding a routing message R: if each intermediate node has to buffer message R for a key valid time before message R is forwarded, the end to end forwarding delay will accumulate in each hop such that the delay will become very large and significantly decrease the routing message forwarding performance. Fortunately, DRETA buffers the forwarded message R while the originator of message R broadcasts the message. Fig. 3 shows that the originator A buffers message R, signs R with Ki as R's Validation Message (VM), and broadcasts R's VM. Then, intermediate node B forwards R's VM immediately to C as node B would forward the message R. Node B and C both have R's VM in front; because of this, while the originator A floods message R, node B and C can validate message R and forward R instantly. Thus, DRETA waits for one key valid time while flooding a forwarded message instead of accumulating large end to end delays [5].

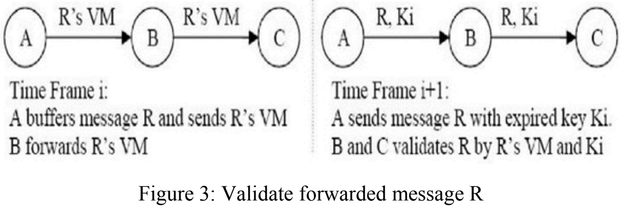
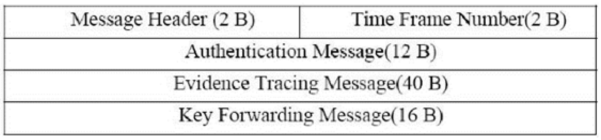

A
R's VM
B
R's VM
C
Time Frame i:
A buffers message R and sends R's VM
B forwards R's VM
A
R, Ki
B
R, Ki
C
Time Frame i+1:
A sends message R with expired key Ki.
B and C validates R by R's VM and Ki

Figure 3: Validate forwarded message R

C. VALIDATION MESSAGE

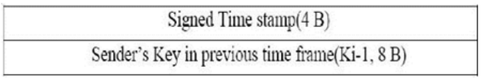


Message Header (2 B)	Time Frame Number(2 B)
Authentication Message(12 B)	
Evidence Tracing Message(40 B)	
Key Forwarding Message(16 B)	

Figure 4: Validation Message Format in Byte (B)

DRETA proposes the utilization of Validation Messages to authenticate routing messages and follow routing proof of forwarded routing messages. In Fig. 4, a Validation Message comprises of three sub-messages: An Authentication Message, an Evidence Tracing message, and a Key Forwarding Message. Since authentication administration is key, a Validation Message must contain an Authentication Message. A Validation message contains an Evidence Tracing message just when the relating routing message is a forwarded message, and Key Forwarding Message goes with an Evidence Tracing message [9].

D. AUTHENTICATION MESSAGE



Signed Time stamp(4 B)
Sender's Key in previous time frame(Ki-1, 8 B)

Figure 5: Authentication Message

In Fig. 5, an Authentication Message comprises of the last sender's lapsed key and the marked expecting sending time of the buffered routing message. Initially, in a timeframe i, the last terminated key Ki-1 is utilized to approve the late gotten routing message, which is marked by Ki-1. The key Ki-1 additionally works as the public key to accept the up and coming Ki. Since the current key Ki is the private key of the sender, the key likewise works as an authentication personality for routing messages [12].

Second, the marked expected sending time of the routing message gives non-notoriety and forestalls replay attacks.

E. BOOTSTRAPPING

Because keys of nodes are valid for a particular time interval, time synchronization is required among the nodes. We assume that nodes register themselves with a shared trusted server before joining the network. When they register, they can obtain their DNS, DHCP, and time synchronization service. Time synchronization error should be less than transmission time to prevent replay attacks. They also obtain their public and private key, a certificate of their public key, and the server's public key for bootstrapping. When a node reboots and starts to release the first key of its key chain, it signs the first key using its private key. Then it releases the first key along with the digital signature of the first key, its public key, and certificate of the public key to all the other nodes.

F. TRACING FORWARDED ROUTING EVIDENCE AND PREVIOUS FORWARDER

DRETA utilizes key-Hashed Message Authentication Code (HMAC) to ensure the integrity of flooding routing messages. For instance, a message M is flooded and the forwarding route is [N0 — * ... — * Ni −1 — * Ni — * Ni + 1 — * ... — * Kn]. Each node in the route signs the hash value of message M as its HMAC and advances its HMAC. Subsequent to sending its HMAC, the node advances message M; every hub has the HMAC of the sender to approve the integrity of M.

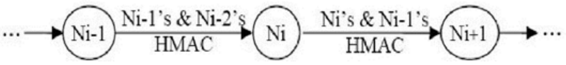
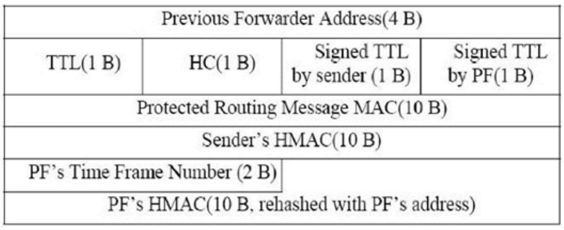


Figure 6: Sending HMACs of Sender and Previous Forwarder (PF)

Nonetheless, the sender Ni might noxiously alter the substance of M. In this way, DRETA requires that the sender Ni additionally advances the past sender's (Ni-1's) HMAC together with sender's (Ni's) HMAC, as appeared in Fig. 6. The past sender, Ni-1, is known as the "Previous Forwarder", which is the forwarder who forwarded M to the sender Ni. At that point, the receiver, Ni+1, can have HMACs from both Ni and Ni-1, and node Ni+1 can accept M from Ni-1's HMAC. In this way, DRETA can follow the forwarded routing evidence utilizing the HMAC of the Previous Forwarder (PF) and guarantee the integrity of the forwarded messages [12].

G. EVIDENCE TRACING MESSAGE



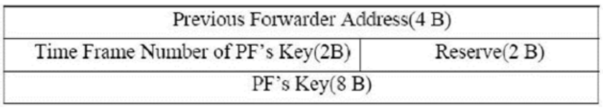
Previous Forwarder Address(4 B)			
TTL(1 B)	HC(1 B)	Signed TTL by sender (1 B)	Signed TTL by PF(1 B)
Protected Routing Message MAC(10 B)			
Sender's HMAC(10 B)			
PF's Time Frame Number (2 B)			
PF's HMAC(10 B, rehashed with PF's address)			

Figure 7: Evidence Tracing Message

In Fig. 7, an Evidence Tracing Message (ETM) conveys the Message Authentication Codex (MAC) of the up and coming forwarded routing message from originator and the relating HMACs of the sender and Previous Forwarder (PF). Each forwarding node utilizes a MAC to produce its HMAC, and stores the sender's and PF's HMAC to approve the up and coming forwarded message. At that point the node forwards another ETM for the up and coming routing message if the node should forward the message. In the new ETM, the sender's HMAC is marked by the node, and PF's HMAC is duplicated from the sender's HMAC utilizing the already gotten ETM. Keeping in mind the end goal to demonstrate that PF's HMAC is sent by the sender and keep the receiver from reusing the PF's MAC, the sender rehashes PF's HMAC with the sender's address and replaces PF's HMAC. Finally, the receiver will have PF's key from the Key Forwarding Message (KFM) with a specific end goal to re-make the PF's HMAC [14].

Variable fields, for example, Time olive (TTL) and Hop Count (HC), are basic and can't be secured by HMAC on the grounds that they overhaul as the routing message is sent one hop further. Along these lines, ETM likewise conveys TTL and HC, and the scrambled TTL of the sender and previous forwarder (PF). At that point the beneficiary can approve sender's TTL utilizing the PF's TTL (PF's TTL = sender's TTL + 1). Since MAC contains TTL originator's TTL and MAC is confirmed by PF's HMAC, sender's HC can be checked by originator's and PF's TTLs (PF's HC = originator's TTL - PF's TTL; sender's HC = PF's HC + 1). In this manner, DRETA can utilize Evidence Tracing Message to trace and ensure the integrity of the routing evidence in forwarded routing messages [9].

H. KEY FORWARDING MESSAGE



Previous Forwarder Address(4 B)	
Time Frame Number of PF's Key(2B)	Reserve(2 B)
PF's Key(8 B)	

Figure 8: Key Forwarding Message

In Fig. 8, a Key Forwarding Message (KFM) conveys the last Previous Forwarder's lapsed key. A KFM is sent with the relating routing message so that the receiver can quickly approve the routing message with an ETM and the keys in a KFM. The basic mix of messages in a Validation message incorporates the accompanying: an AM and an ETM for an up and coming forwarded routing message and a KFM for the as of now forwarded message. What's more, a forwarded routing message is sent with the Validation Message. Note that the key contains the identity number, demonstrating the position of the key in the keychain. Consequently, the receiver can approve the key utilizing the previous key alongside the keychain [7].

I. DRETA FINITE STATE MACHINE

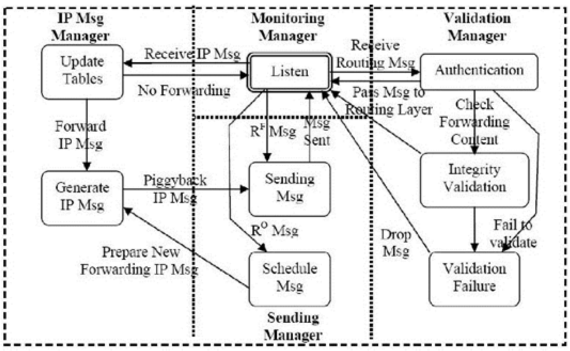


Figure 9: DRETA Finite State Machine within a node

R' Msg: Outgoing Forwarded Routing Message (Sender is Not Originator)

Ro Msg: Outgoing Originated Routing Message (Sender is Originator)

In Fig. 9, DRETA consists of four component ID Managers running at each node. The Monitor Manager intercepts incoming and outgoing routing messages and handles Validation Messages [5].

The Evidence Manager records evidence stored in the incoming Validation Messages. The evidence will be used to validate the corresponding routing message, such as expected sending time in AM, PF's HMAC in ETM, and sender's and PF's keys.

The Sending Manager sends outgoing routing and validation messages. If the routing message is a forwarded message (R′ Msg), he manager forwards the message immediately. If the message originates from the node (R° Msg), the manager buffers the message and schedules the sending time, which is after the current key expires [7].

Also, it generates a new VM by using the current key for the buffered routing message, and sends the new VM. Then, the manager sends the buffered message at the scheduled time.

The Validation Manager validates the received routing messages. The manager first authenticates it according to the evidence given in the previously received AM. If the routing message is a forwarded message, the manager validates its content using evidence in the received ETM. If the validation fails, the message is dropped. Otherwise, it is passed on to the routing layer.

V. DRETA IMPLEMENTATIONS

Here we discuss implementations of DRETA with AODV, OLSR, and DEMEM. AODV is a common and famous request-on-demand protocol. Like other request on demand protocols, AODV heavily relies on forwarded messages, so DRETA is an ideal solution to protect AODV. OLSR is the other typical proactive protocol in MANETs. OLSR has two essential routing messages, Hello and TC messages. DRETA can protect the content of forwarded TC messages, and DEMEM protects the initiated Hello and TC message. In addition, DRETA protects DEMEM's ID message in OLSR.

DRETA provides two security protections for AODV, OLSR, and DEMEM. First, DRETA provides authentication service by offering Authentication Messages for those protocols. Second, DRETA provides forwarding protection for their forwarded messages by using Evidence Tracing Messages.

A. DRETA in AODV

DRETA has two security restrictions to prevent AODV messages from being maliciously modified when they are created (that is, the message originator is the malicious node). First, DRETA does not allow an intermediate node to reply to a RREP because [13] the intermediate node is not the originator of RREP. In addition, the routing data of the destination in the intermediate node may be outdated, and it is difficult and expensive to validate the routing data. Therefore, it is much safer to only allow the destination to reply a RREP. Second, nodes ignore Sequence Numbers of the destination in RREQ and RERR because the number may also be outdated and therefore not trustworthy [10].

In other situations, if the originator provides incorrect information in their AODV messages, the incorrect information will only cause routing damage for the originator itself. For example, if the originator provides decreasing Sequence Number (SN) in its created AODV message, the message will be ignored since the SNs are smaller. Since the originator is the only one that can increase its own SN, if the originator increases its SN a large amount, it will not affect AODV operation. Thus, attackers cannot benefit from malicious originated AODV messages. Because DRETA has secured forwarded AODV messages and authenticated all AODV messages, DRETA successfully protects the integrity of AODV messages.

B. DRETA in OLSR

OLSR has two main routing messages, non-forwarded Hello messages and forwarded TC messages. DRETA provides authentication for all messages, and provides forwarded message protection for TC message. In OLSR, only MPR nodes forward TC messages, so Evidence Tracing Messages (ETM) and Key Forwarding Messages (KFM) are only forwarded by MPR nodes; ETM carries the HMAC of the TC message to protect the TC message. Again, TTL and HC are secured separately in ETM, since abnormally large or small TTLs may still cause problems. Thus, DRETA secures the forwarded TC messages in OLSR and prevents attacks using attack method 4 in Fig. 1.

DEMEM has three ID messages: ID-Evidence, ID-Forward, and ID-Request, to protect the integrity of the Hello and originated TC messages in OLSR. DEMEM prevents attacks using one of the first three attack methods in Fig. 1. DRETA also authenticates three ID messages. Since the ID-Evidence message is a forwarded message, DRETA protects the integrity of the ID-Evidence message. Therefore, DRETA and DEMEM cooperatively ensure the integrity of the routing messages in OLSR [7].

VI. EXPERIMENT

We implemented DRETA in GloMoSim, a simulation designed for MANETs. First, we introduce the experimental environment. Then, we present the experimental results with three performance metrics.

A. EXPERIMENT ENVIRONMENT

GloMoSim is a straightforward, compelling, and adaptable trial reproduction stage intended for MANETs. GloMoSim underpins 802.11, different routing protocols in MANETs, (for example, AODV and OLSR), and Ground Reflection (Two-Ray) radio model.

The radio model has both a direct path and a ground reflected propagation path between the transmitter and receiver. Here, the radio range is around 377 meters, calculated with the accompanying parameters [9]: antenna height of 150cm, transmission power of 15dBm, antenna gain of 0, sensitivity of - 91 dBm, and receiving threshold of - 81 dBm.

Network topologies comprise of 15 and 25 nodes in 1km x 1km, 50 nodes in 1.5km x 1.5km, 100 nodes in 2km x 2km, and 150 nodes in 2.5km x 2.5km. In these five network topologies, nodes are arbitrarily put in the just as separated cells in the field. Mobile nodes take after the Random way point Mobility Model with random speeds of up to 20 meters/sec (45 miles/hr.), and node pause times change from 0 to 300 seconds. Absolute simulation time is 600 seconds [9]. DRETA utilizes the SHA-1 hash function to create MACs and HMACs. The hash value size is 10 bytes and the key size is 8 bytes. The key expire time is 1 second.

B. PERFORMANCE METRICS

First, we define three performance metrics to measure DRETA's overhead: 1) message overhead, 2) detection accuracy, and 3) routing message delay. Second, the experiment is performed under two kinds of testing conditions: mobility and scalability.

For mobility, node speed can be 0-10 (low) or 1-20 (high) m/s, and node pause times are 0, 30, 60, 120, 180 and 300 seconds. For scalability, the number of nodes can be 25, 50, 100 and 150. Mobility is tested in 50 nodes, and scalability is tested under the highest mobility (1-20 m/s, no pause time).

The metrics are used to measure DRETA's overhead in AODV or OLSR together with DEMEM. Since DRETA uses symmetric keys, DRETA's computation overhead is close to regular routing protocols [11].

Message Overhead

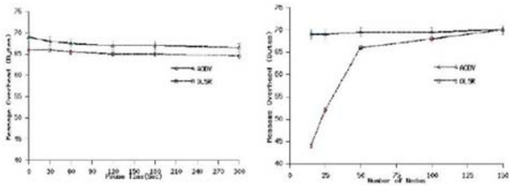


Figure 10: Message Overhead (Average Validation Message size in Bytes)

Since a routing message requires one Validation Message (VM) on average, message overhead is considered normal Validation Message size. Forwarded messages require extra Evidence Tracing Message (ETM) and Key Forwarding Message (KFM), yet non-sent messages just need Authentication Message (AM). Average Validation Messagesize is 16 bytes (AM size) + 56 bytes (ETM + KFM size) × forwarded message ratio (ETM+KFM frequency ratio) so its greatest is 72 bytes (when forwarded message ratio is 100%). Accordingly, the forwarded message ratio is basic for figuring validation message size [15].

For mobility, Fig. 10 shows that mobility has little effect on message overhead, since mobility does not affect the forwarded message ratio. In AODV, mobility has a higher influence because higher mobility results in more lost links and more RREQ and RREP. In OLSR, lost links have little effect on the ratio of forwarded messages because periodic routing messages (e.g. TC messages) will be reflected in the broken links. Therefore, the message overhead changes very slightly as mobility increases.

For scalability, in AODV, the forwarded message ratio is high and stable so that the message overhead remains the same as the number of nodes increases. In OLSR, the ratio of forwarded TC messages increases as the number of nodes increases, so message overhead increases, but does not exceed 72 bytes. In addition, message overhead in OLSR is lower than that of AODV on average. In general, DRETA is scalable for message overhead [8].

Detection Accuracy

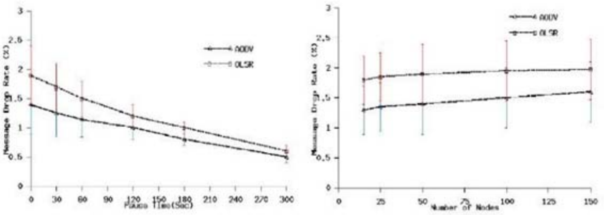


Figure 11: Routing Message Drop Rate

For a broadcast forwarded message, its Validation Message is a broadcast message. If this broadcast VM is lost because of lost links or message collision, the routing message will be dropped due to lack of a corresponding VM and may lead to a false positive. Fig. 11 shows that in most cases, routing message drop rate (false positives) is low. For mobility, the rate is lower as mobility decreases. For scalability, the rate slightly increases as the number of nodes increases. The rate in OLSR is slightly higher than that in AODV because the rate usually results from lost links, and lost links in AODV expire faster than in OLSR. No false negatives occur in AODV and OLSR. Since the number of nodes has little effect on the accuracy, DRETA is scalable for detection accuracy. Thus, DRETA has great accuracy in AODV and OLSR [5].

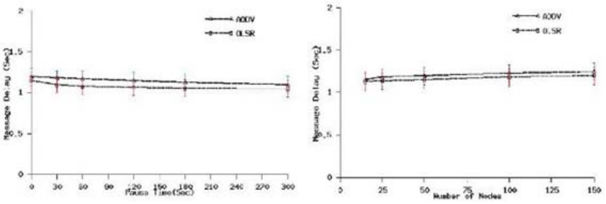


Figure 12: Routing Message

Delay Routing Message Delay

The delay mainly results from the delay key disclosure; the key interval is one second, so the delay is a little bit more than a second. Fig. 12 shows that the mobility and the number of nodes have very little influence for the message delay, so the delay time is usually very stable.

VII. CONCLUSION

DRETA is a scalable, efficient, and generic intrusion detection model for protecting and authenticating forwarded routing messages in MANETs. First, DRETA adopts symmetric keys and HMAC for low computation overhead message integrity protection, and adopts one-way key chaining and delay key disclosure for public key style key management. Second, DRETA offers a distributed intrusion detection architecture. Because DRETA intercepts routing messages within a node and functions separately from routing protocols, DRETA is a scalable and generic intrusion detection model for MANETs. Third, DRETA proposes Validation Messages, which carry keys and HMACs for authentication and forwarded message integrity protection. In addition, DRETA proposes a Previous Forwarder to propagate Validation Messages with minimal message overhead and to perform scalable routing evidence tracing. Fourth, we implement DRETA in two representative routing protocols, AODV and OLSR, in MANET. DRETA successfully protects AODV message integrity and protects OLSR message integrity in cooperation with DEMEM. Last, experimental results in GloMoSim show that DRETA instantly protects AODV and OLSR messages under high mobility situations and has low computation and message overhead, low false positives, no false negatives, and low packet delay. The limitation in DETRA is that DRETA cannot detect correlated attacks, such as tunneling attack.

REFERENCES

[1] Pai, Ganesh J. "A survey of software reliability models." arXiv preprint arXiv:1304.4539 (2013).

[2] A. Dvir B. Ben-Moshe, E. Berliner and A. Gorodischer. A joint framework of passive monitoring system for complex wireless networks. In Workshop on Emerging Densely Connected Network, IEEE Consumer Communications and Networking Conference, Navada, USA, Jan. 2011.

[3] M. Baker, K. Giuli, and S. Marti. Mitigating routing misbehavior in mobile ad hoc networks. In INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, pages 255–265. ACM,2000.

[4] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. Personal Communications, IEEE,7(5):28–34, 2000.

[5] L. Chen, S.H. Low, M. Chiang, and J.C. Doyle. Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. InInfocom, 2006.

[6] B.P. Crow, I. Widjaja, LG Kim, and P.T. Sakai. Ieee 802.11 wirelesslocal area networks. Communications Magazine, 35(9):116–126, 1997.

[7] P. Gardner-Stephen. The serval project: Practical wireless ad-hoc mobile telecommunications, 2011.

[8] Paul Gardner-Stephen and Swapna Palaniswamy. Serval mesh software wifi multi model management. In Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief, pages71–77, Kerala, India, 2011.

[9] Z. Hua. A survey of quality of service in ieee 802.11 networks. IEEE Wireless Communications, 11:6–14, 2004.

[10] [K. Jain, J. Padhye, V.N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. Wireless networks,11(4):471–487, 2005.

[11] A. P. Jardosh. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis. IEEE,2005.

[12] J. M. M. Kamal, M. S. Hasan, A. L. Carrington, and Y. Hongnian. Lessons learned from real manet experiments and simulation-based evaluation of udp and tcp. In International Conference on Computer and Information Technology, pages 511–515, Bangladesh, India, Dec.2010.

[13] H. Kazemi, G. Hadjichristofi, and L. A. Da Silva. MMAN - a monitor for mobile ad hoc networks: design, implementation, and experimental evaluation. In International workshop on Wireless network testbeds,experimental evaluation and characterization, pages 57–64. ACM, 2008.

[14] S. Khurana, A. Kahol, and A. P. Jayasumana. Effect of hidden terminals on the performance of ieee 802.11 mac protocol. In IEEE LocalComputer Networks, 1998.

[15] N. R. Krishna, M. B. Elizabeth, and C. A. Kevin. DAMON: A distributed architecture for monitoring multi-hop mobile networks. In Proceedings of IEEE SECON, 2004.