MOBiWORP: Mitigation of the Wormhole Attacks in Mobile Multi-hop Wireless Networks (MANET)

Issa Khalil, Saurabh Bagchi, and Ness B. Shroff

Dependable Computing Systems Lab (DCSL) & Center for Wireless Systems and Applications (CWSA)
School of Electrical and Computer Engineering
Purdue University

Outline

• Introduction
  – What is a MANET network?
  – Attacks against MANET networks
  – The wormhole attack
  – The goals of the paper

• Primitive Building Blocks
• Mitigating Wormhole Attack in Mobile Networks
• Conclusion
MANET networks

- Mobile Ad Hoc NETworks (MANET)
- Autonomous system of nodes with no static infrastructure
- All or subset of nodes may move
- Nodes communicate wirelessly in multi-hop fashion
- Often subject to rapid deployment in environments where natural or malicious errors are likely

Security Attacks

Two classes of attacks

1. Attacks that can be defeated by crypto mechanisms
   - Eavesdropping: Solved by encryption
   - Message tampering: Solved by authentication

2. Attacks that subvert the functionality of the network
   - Control attacks: Manipulating control traffic (e.g., routing traffic) to disrupt data traffic
     - Examples: ID spoofing and Sybil, sinkhole, rushing, wormhole
   - Data Attacks: Directly manipulate data traffic
     - Examples: Blackhole, grayhole
   - This class cannot be prevented by cryptographic mechanisms alone
What is the Wormhole Attack?

- A control traffic attack that enables an attacker node to draw many routes through it
- Attacker tunnels packets received in one part of the network and replays in another part giving the illusion that optimal routes pass through it
- Tunneled packets look legitimate thus crypto mechanisms cannot detect them
- Puts the attacker in a powerful position to disrupt network functionality
  - Insinuate attacker in a route and then manipulate data traffic
    - Example: Selectively drop data packets
  - Routing disruptions
    - Example: Prevent discovery of legitimate route
  - Traffic analysis
    - Example: Observe traffic patterns as a way of leaking information
- Particularly insidious because can be launched without possessing any cryptographic keys

How can a Wormhole Attack be Launched?

- A simple way to launch the wormhole attack is through an out-of-band channel [1]
- Collusion is required for the attack to succeed

- S-C-B-E-D is a 4-hop legitimate route
- S-X-Y-D is a 3-hop wormhole route

Goals

- Mitigate the wormhole attack in MANET networks with mobile attacker by
  - Detecting nodes involved in the attack
  - Diagnosing attacker nodes
  - Isolating attacker nodes from the network

- All previous approaches
  - Either, use expensive hardware, such as tight time synchronization among all nodes, directional antennas, etc.
  - Or, rely on all nodes being static and therefore their neighbors are unchanging

Outline

- Introduction
- The Wormhole Attack
- Primitive Building Blocks
  - Local Monitoring
  - Detection using local monitoring
  - Why detection is imperfect
- Mitigating Wormhole Attack in Mobile Networks
- Conclusion
Local Monitoring

- A collaborative detection strategy in which a node monitors the traffic going in and out of its neighbors.
- Assumptions
  - Each node knows its first-hop and second-hop neighbors
  - Requires each node to include the ID of the prev-hop in the forwarded packet
- A guard of a node A for the link from X to A is any node that lies within the transmission range of both X and A
  - Example: M, X, and N are the guards of node A for the link from X to A
- A guard saves information about incoming packets in a watch buffer
- Matches an output packet with information in buffer

Local monitoring: Details

- Local monitoring can be used to detect different kinds of control attacks by changing the information maintained in the buffer and the kind of checking that goes on
- The different kinds of malicious activity that can be done by a node
  - Fabrication
  - Modification
  - Delay
  - Drop
- Correspondingly the kind of checking that needs to be done are:
  - An outgoing packet that has no matching packet in watch buffer
  - Difference between incoming and outgoing packet fields
  - Forwards after a threshold time
  - Not forwarding within a maximum acceptable timeout threshold
Detection Using Local Monitoring

Attacker goal: including malicious nodes in the route

Choice#1
M1 claims that the RREP is from M2, from off R of its neighbors, say Z

Detection strategy
The guards of M1 over the link ZÆM1, (P,Z,Q) detect this malicious behavior, because they have nothing in their watch buffers about RREP coming from Z

Choice#2
M1 claims that the RREP comes from one of its neighbors, say Z

All the neighbors of M1 (S,R,P,Z,Q,B) detect this malicious activity, because they know that M2 is not a neighbor of M1

Why Detection is Imperfect

Due to collision the following may occur

• *Missed detection*: A malicious event goes undetected
  – Collision at the guard (G) when the node (D) forwards a packet
• *False detection*: A normal event is detected as a malicious event
  – Collision at the guard (G) when the sender (S) transmits a packet
  – Detection at the guard when the monitored node (D) forwards the packet
Outline

- Introduction
- Primitive Building Blocks
  - Mitigating Wormhole Attack in Mobile Networks
    - The mobility challenge
    - System assumptions and attack model
    - Data structures
    - The selfish move protocol (SMP)
    - The connectivity-aided protocol with constant velocity (CAP_CV)
    - Detection & isolation of malicious nodes
    - Results
- Conclusion

The Mobility Challenge

- No fixed neighborhood membership
- Need two-hop neighbor verification that is
  - Efficient in time and energy
  - Secure
  - Not relying on expensive hardware
- No existing solution satisfies these requirements
- In MOBIWORM, we provide
  - Two-hop neighbor verification whenever there is the possibility of launching a wormhole attack
  - Use this information to mitigate the wormhole attack with mobile attackers
Assumptions & Model

- **System assumptions**
  - Existing key distribution mechanism
  - Mix of mobile and static nodes
  - Bi-directional links
  - Network has unconstrained trusted central authority (CA)
  - Ability to verify CA signatures
  - The maximum number of nodes in the network that can be compromised is known a priori
  - Loose time synchronization

- **Attack model**
  - Links may be subjected to eavesdropping and message tampering
  - Attacker node may be external or internal (i.e., compromised node)
  - Attacker node may be more powerful than legitimate network nodes
  - Attacker can arbitrarily delay, drop, modify, or fabricate subset of packets
  - Attacker nodes can collude among one another
  - Brute force denial of service attacks are not considered

Data Structures

- A node $B$ maintains
  - MalC($B$,i) about each neighbor $i$
  - Neighbor list ($N_{B_list}$)
  - Black List ($B_{list}$) of known revoked nodes

- The CA suspicion table ($ST_{globe}$)
  - $N+1 \times N$ table, $N =$ number of nodes
  - $ST_{globe}[i,j].sf = 1$ if $i$ revokes $j$, 0 otherwise
  - $ST_{globe}[i,j].ctr = MalC(i,j)$
  - $ST_{globe}[i,j].Time =$ the aggregated continuous time during which $i$ & $j$ are neighbors
  - $ST_{globe}[N+1,j].sf = 1$ if $j$ has been globally revoked
  - $ST_{globe}[N+1,j].ctr =$ number of nodes locally revoke $j$
ANUM & Node States

- **Authentication Neighbor Update Message (ANUM)**
  - A certificate given by CA to a node (signed using CA’s private key)
  - Used to convince other nodes of location ANUM.Loc
  - Has an expiration time ANUM.T_expire
- **Grace Period (T_grace)**: the max time a node can send and recv after the expiration of its ANUM
- **Node States**: based on the functionality allowed to the node
  - **Valid (send, recv, relay)**
    - Claimed Loc = ANUM.Loc and
    - Cur_Time < ANUM.T_expire
  - **Incorrect (send, recv)**
    - Claimed Loc != ANUM.Loc or
    - ANUM.T_expire < Cur_Time < T_grace
  - **Invalid (only Handshake packets)**
    - Cur_Time > T_grace
  - **Revoked (no allowable functionality)**

---

Selfish Move Protocol (SMP): Getting ANUM

Used for scenarios in which a mobile node is not allowed to relay packets while moving

1. **ANUM_Req**
   - Encrypted using shared key
   - Includes new location and pause time at that location
   - Signed by CA
   - Each node that overhears ANUM_Reject, adds B to its BList
2. **ANUM_Reject**
   - If B has a valid ANUM, CA drops the ANUM_Req
   - If B is revoked, CA sends ANUM_Reject
   - If B has no valid ANUM and not revoked, CA sends ANUM_Rep
3. **ANUM_Rep**
   - Signed by the CA
   - Includes: B, new location, expiration time
**SMP: Using the ANUM**

Initiated by a node when it reaches its new location

1. If W in incorrect state, W marked in Nblist(B)
2. Add B to Nblist
3. ANUM(W), Blist(W)
4. Blist(W) is authenticated using the shared key
5. One-hop broadcast
   - Blist(B) authenticated using the shared key

• A malicious node directly detected in Blist(B) serves as a partial detection evidence to W and vice-versa
• At min(Texpire(W), Texpire(B)), B removes W from Nblist(B), so does W

**Connectivity Aided protocol with Constant Velocity (CAP_CV)**

- Problems with SMP
  - Network may get disconnected in high mobility scenario
  - A moving node can not communicate beyond T_{grace}

- Goals
  - Preserve the same network connectivity conditions as the insecure network
  - Allow moving nodes to travel any distance

- Assumptions
  - Each mobile node knows its location and trajectory of motion
  - Moving with fixed velocity (v)
**CAP.CV: Getting and Using the ANUM**

- Getting the ANUM (retry if fail)
  - Encrypted using shared key
  - Includes \((X_0, Y_0, X_1, Y_1, T_{start}, v)\)
  - Signed by CA
  - Each node that overhears ANUM_Reject, adds B to its BList

- If B has a valid ANUM, CA drops the ANUM_Req
- If B is revoked, CA sends ANUM_Reject
- If B has no valid ANUM and not revoked, CA sends ANUM_Rep

- Secure neighbor discovery: same as SMP except that B computes the difference between actual position and computed one and refrain from broadcast if greater than a threshold

**Local Isolation**

- Goals
  - Propagate detection knowledge among the first-hop neighbors of the attacker
  - Isolate the malicious node from its local neighborhood
- When a guard G detects a malicious event by node M
  - G increments \(\text{MalC}(M,G)\)
  - Different malicious activities can be considered at different levels of criticality
- When the \(\text{MalC}(M,G)\) crosses a threshold
  - G removes M from its neighbor list
  - G sends an authenticated alert to the neighbors of M
- When W receives an alert about a neighbor M
  - Collects alert information from multiple guards
  - When the number of alerts reaches detection threshold \(\gamma\), W removes M from its neighbor list
- Local isolation is not sufficient for mobile attacker nodes
  - A malicious node leaves the current neighborhood to a new one
Global Isolation

- Upon detection of a malicious node, M, a guard G sends an alert to the CA
  - I directly detect M behaving maliciously, or
  - This is the MalC(G,M) and the length of the monitoring round
- The CA updates its data structure (ST\textsubscript{globe}) accordingly
  - ST\textsubscript{globe}[G,M].s_{r} = 1, ST\textsubscript{globe}[N+1,M]++ or
  - Update ST\textsubscript{globe}[G,M].Ctr and ST\textsubscript{globe}[G,M].Time
- CA takes decision about M
  - If ST\textsubscript{globe}[N+1,M] = the bound on the number of compromised nodes + false alarm safety factor, mark M as malicious

Simulation Setup

- Use ns-2 network simulator
- Data communication model: any node to any node, uses AODV for routing
- Node distribution: Randomly on a fixed-size field
  - Increasing number of nodes increases the density
- Node movement: Random way-point model with velocity picked randomly from a uniform distribution (v_{min}, v_{max})
- Attacker nodes: Internal compromised nodes randomly selected from network
- Attack model for wormhole attack: Out-of-band channel emulated by allowing instantaneous packet forwarding among attacker nodes
  - Attacker nodes drop all data packets through them
  - Attacker nodes have perfect collusion
- We simulate two scenarios
  - Baseline: insecure network
  - MobiWorp
- The output parameters are measured at the end of simulation time (1500 s)
**Results: Drop Ratio**

The output here is Drop ratio = % (Packets dropped/Packets sent)

![Graph showing drop ratio over simulation time]

- Drop ratio in Baseline is higher and reaches a steady state with time
- In MobiWorp, drop ratio goes to zero with time due to isolation
- The higher the number of nodes, the smaller is the fraction of malicious nodes and therefore the lower the drop ratio

**Input parameters**
- $\gamma = 3$
- # mal. = 4
- #nodes= 80,90,100

---

**Results: Local Isolation**

- % Isolation = %(#mal. nodes isolated locally/total number of mal. nodes)
- % False isolation = %(#good nodes isolated locally/total number of nodes)

![Graph showing local isolation and false isolation]

- Both isolation and false isolation decrease with increasing $\gamma$ since it becomes more difficult to get consensus among guards
- Why do we go for infinite $\gamma$ if $\gamma = 4$ is good enough?

**Input Parameters**
- # mal. Nodes = 4
- # of nodes = 60
Results: Global Properties

- %Global Isolation = % (the # mal. nodes revoked by CA / total of mal. nodes)
- %Global False Isolation = % (the # nodes falsely revoked by CA / total # nodes)
- %Global Isolation time = average of global isolation time of each isolated malicious node
- Isolation time of a malicious node = the time from which the node starts attacking the network to the time when the node is revoked by the CA.

- Low γ requires contribution of many neighborhoods and thus low isolation and false isolation percentages
- Global latency decreases even though local latency increases with γ

Results: Effect of Motion

Duty cycle = motion time / simulation time

- Increasing frequency of motion causes malicious nodes to escape before the MalC reaches the threshold. CA does not aggregate across guards. This decreases both detection and false detection
- In Base case increasing motion frequency causes wormholes to break faster and thus the drop ratio decreases

Input parameters
- γ = 3
- #Max mal. = 3
- # mal. = 4
- #nodes= 60
Conclusion

• Proposed a generic strategy for cooperative distributed detection of the wormhole attack in mobile ad-hoc networks (MOBIWORP)
• Proposed a generic strategy for locally isolating the malicious nodes
• Proposed a global strategy for mitigating the wormhole in face of mobile malicious nodes through the CA
• Study the efficiency of MOBIWORP under different network conditions and mobility patterns
• Future Work:
  – Extension to aggregate across multiple guards
  – Scheduling of guards to reduce collision

Thanks

Questions?