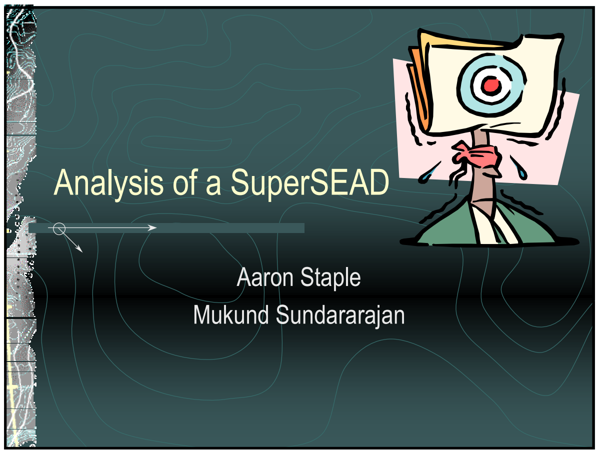
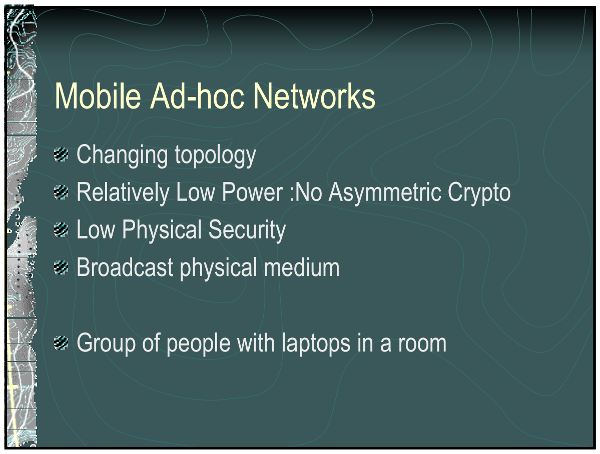


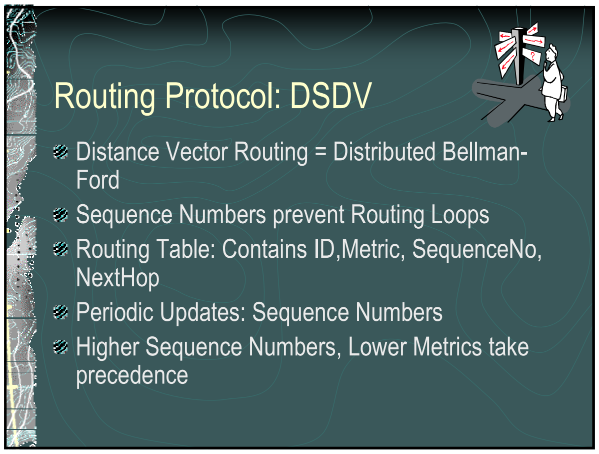
Analysis of a SuperSEAD

Aaron Staple
Mukund Sundararajan



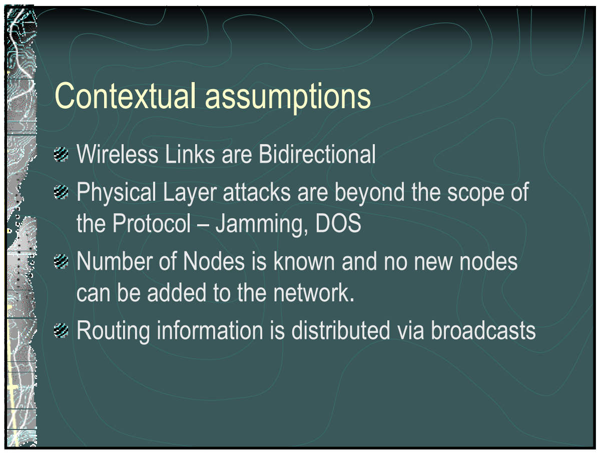
Mobile Ad-hoc Networks

- Changing topology
- Relatively Low Power :No Asymmetric Crypto
- Low Physical Security
- Broadcast physical medium
- Group of people with laptops in a room



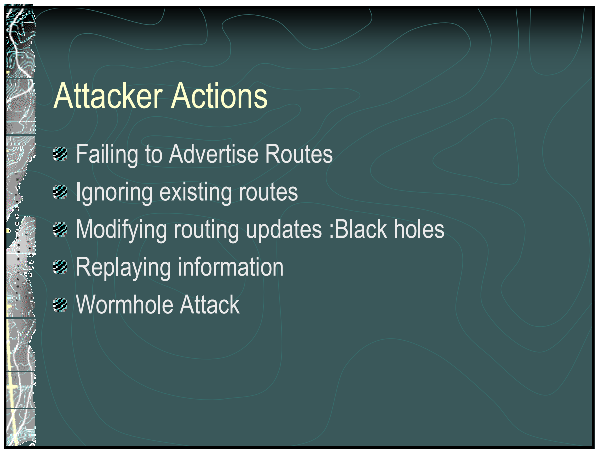
Routing Protocol: DSDV

- Distance Vector Routing = Distributed Bellman-Ford
- Sequence Numbers prevent Routing Loops
- Routing Table: Contains ID,Metric, SequenceNo, NextHop
- Periodic Updates: Sequence Numbers
- Higher Sequence Numbers, Lower Metrics take precedence



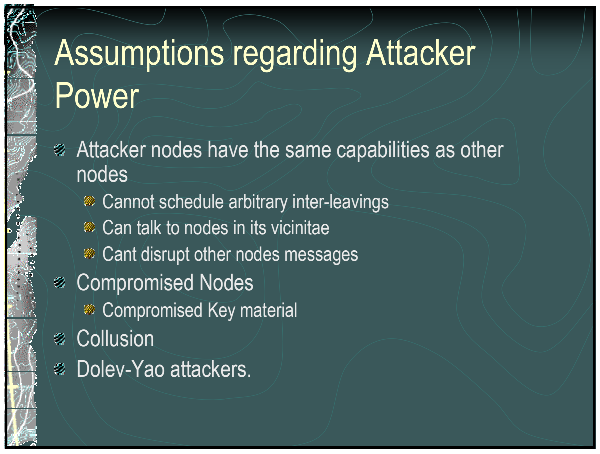
Contextual assumptions

- Wireless Links are Bidirectional
- Physical Layer attacks are beyond the scope of the Protocol – Jamming, DOS
- Number of Nodes is known and no new nodes can be added to the network.
- Routing information is distributed via broadcasts



Attacker Actions

- Failing to Advertise Routes
- Ignoring existing routes
- Modifying routing updates :Black holes
- Replaying information
- Wormhole Attack



Assumptions regarding Attacker Power

- Attacker nodes have the same capabilities as other nodes
  - Cannot schedule arbitrary inter-leavings
  - Can talk to nodes in its vicinitae
  - Cant disrupt other nodes messages
- Compromised Nodes
  - Compromised Key material
- Collusion
- Dolev-Yao attackers.

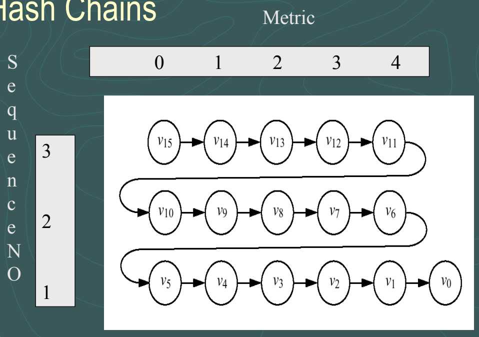
What SuperSEAD attempts to accomplish.

- "SEAD performs well over the range of scenarios we tested,and is robust against multiple uncoordinated attackers creating incorrect routing state in any other node,even in spite of any active attackers or compromised nodes in the network."
- Secure Efficient Ad Hoc Distance Vector Routing

SuperSEAD Protocol

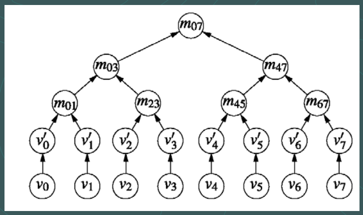
- Hash Tree Chains to Authenticate the lower bound on the metric and an upper bound on the sequence number
- Neighbor Authentication:
  - Origin of the message
  - N^2 Symmetric keys
- Hash Chain Anchors and Symmetric Keys are distributed using an external mechanism
- Packet Leashes :Temporal
  - Avoid replays.

Hash Chains



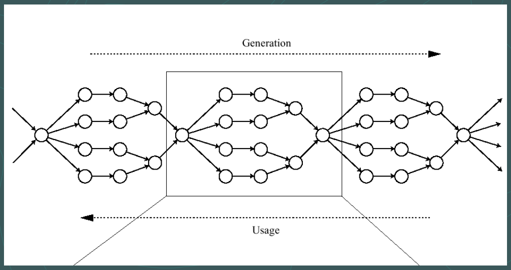
Metric
0 1 2 3 4
Sequence NO
3
2
1
v_{15} v_{14} v_{13} v_{12} v_{11}
v_{10} v_9 v_8 v_7 v_6
v_5 v_4 v_3 v_2 v_1 v_0

Hash Trees

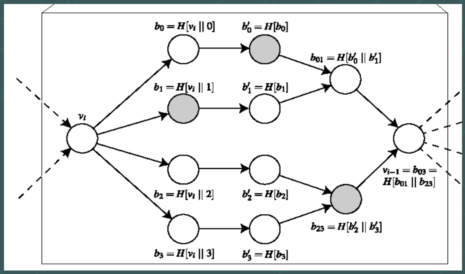


m_{07}
m_{03} m_{47}
m_{01} m_{23} m_{45} m_{67}
v'_0 v'_1 v'_2 v'_3 v'_4 v'_5 v'_6 v'_7
v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7

Hash Tree Chains



Hash Trees Chains Contd..



Analysis Method :Equivalence Based

- Impact on network topology achieved in the presence of a few attacker nodes
- Compare states achievable in the presence of attacker to an attacker-free model
- We consider only steady states
- Essentially simulating the hash chains and the neighbor authentication, assuming that they operate correctly

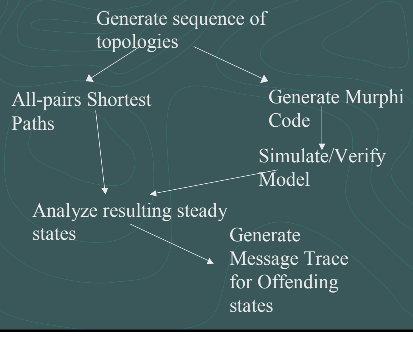
Security Properties

- Correct Routing state at all good nodes about all the other good nodes
- Review of Attacker capabilities:
  - Cant interfere with any messages
  - Cant advertise different information to two different neighbors.
  - Cant perform arbitrary inter-leavings of messages
  - Cannot determine all the events that take place in the network

Murphi Transition Scheduling

- We DO NOT consider all possible schedulings of routing updates
  - Attackers can't control the scheduling
  - Murphi State Space would be extremely large
- We randomly generate permutations that determine order of broadcast updates
- All attacker actions are enumerated

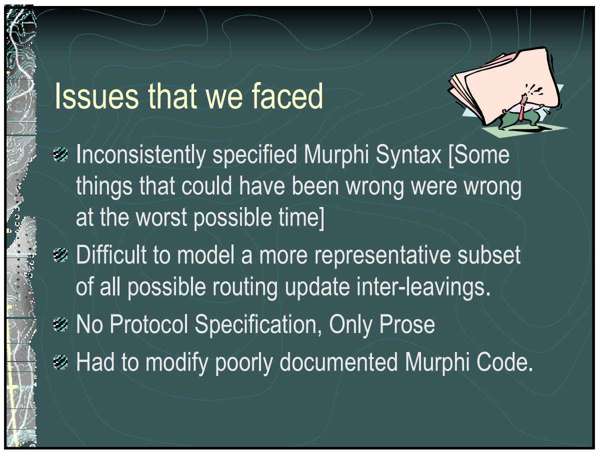
Project Flow


Generate sequence of topologies
All-pairs Shortest Paths
Generate Murphi Code
Simulate/Verify Model
Analyze resulting steady states
Generate Message Trace for Offending states

```
ruleset change: boolean do
ruleset new_seq_no: 0..MaxSequenceNumber do
  ruleset new_distance: 1..(MaxDistance-1) do
    rule 300 "A bad node performs a broadcast update about a single other node"
      (turn_list[turn] > NumGoodNodes)
    ==>
    begin
    for j: 1..TotalNodes do
      if ((topology[top_id][turn_list[turn]][j] = true) &
      ((routing_tables[turn_list[turn]][badAbout].sequence_no > new_seq_no) |
           ((routing_tables[turn_list[turn]][badAbout].sequence_no = new_seq_no) &
           (routing_tables[turn_list[turn]][badAbout].distance + 1 <= new_distance))) &
      ((routing_tables[j][badAbout].sequence_no < new_seq_no) |
           ((routing_tables[j][badAbout].sequence_no = new_seq_no) &
            (routing_tables[j][badAbout].distance > new_distance)))) then

       routing_tables[j][badAbout].sequence_no := new_seq_no;
       routing_tables[j][badAbout].distance := new_distance;
       printout := 1;

      end; end;

    if (badAbout = TotalNodes) then
      turn := (turn % TotalNodes) + 1;
      change_top := change;
    end;
    badAbout := (badAbout % TotalNodes) + 1;

  end; end; end; end;
```

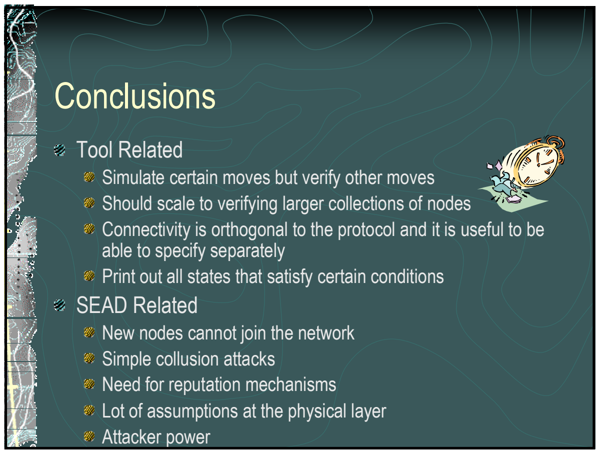
Attacks that we focused on

- Run the protocol without SEAD present and see the attack
- Run it with SEAD present and if an attacker node cannot advertise different information to different neighbors, show that found no attack.
- In the presence of collusion,tunneling there is a wormhole attack.
- Node placement attack.
- K (>1) consecutive attacker nodes on a path can shorten path by k-1.
- Attacks in the absence of neighbor authentication and packet leashes



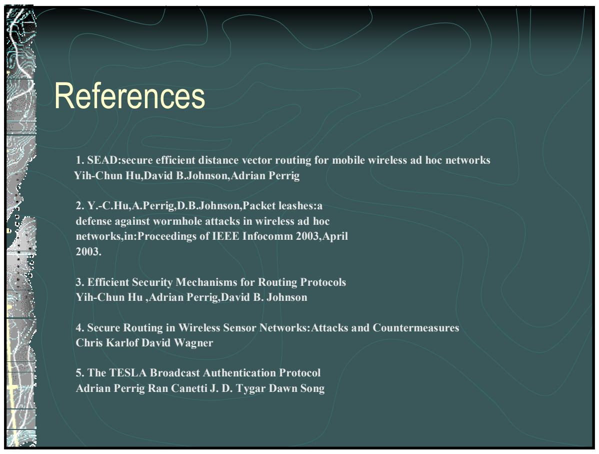
Issues that we faced

- Inconsistently specified Murphi Syntax [Some things that could have been wrong were wrong at the worst possible time]
- Difficult to model a more representative subset of all possible routing update inter-leavings.
- No Protocol Specification, Only Prose
- Had to modify poorly documented Murphi Code.



Conclusions

- Tool Related
  - Simulate certain moves but verify other moves
  - Should scale to verifying larger collections of nodes
  - Connectivity is orthogonal to the protocol and it is useful to be able to specify separately
  - Print out all states that satisfy certain conditions
- SEAD Related
  - New nodes cannot join the network
  - Simple collusion attacks
  - Need for reputation mechanisms
  - Lot of assumptions at the physical layer
  - Attacker power



References

1. SEAD:secure efficient distance vector routing for mobile wireless ad hoc networks
Yih-Chun Hu,David B.Johnson,Adrian Perrig

2. Y.-C.Hu,A.Perrig,D.B.Johnson,Packet leashes:a defense against wormhole attacks in wireless ad hoc networks,in:Proceedings of IEEE Infocomm 2003,April 2003.

3. Efficient Security Mechanisms for Routing Protocols
Yih-Chun Hu ,Adrian Perrig,David B. Johnson

4. Secure Routing in Wireless Sensor Networks:Attacks and Countermeasures
Chris Karlof David Wagner

5. The TESLA Broadcast Authentication Protocol
Adrian Perrig Ran Canetti J. D. Tygar Dawn Song