Secure Routing for Mobile Ad-hoc Networks

Arun Raghavan

Department of Computer Science
IIT Kanpur

CS625: Advanced Computer Networks

Outline

Outline

Need

- Often setting up an infrastructure is infeasible
- Disaster relief
- Community networks (OLPC)
- Military applications
- Enter *MANETs*

Challenges

- No infrastructure
- Wireless (duplicate, delayed packets)
- Mobility
  - Highly dynamic topology
- Devices are usually resource-limited

Outline

Classification

- Table-driven / Proactive
  - Nodes periodically share their routing information with all others
  - Every node has routing information for the entire network
  - Problems w.r.t. efficiency, scalability
  - DSDV, CGSR
- On-demand / Reactive
  - First attempt at making a connection triggers *Route Discovery*
  - Subsequently require *Route Maintenance* in case nodes in a route go down
  - Drawback – setup time for first connection is high
  - AODV, DSR, TORA

Example: DSDV

- Table-driven protocol
- Remember *Distance Vector* routing?
  - And the count-to-infinity problem?
- DSDV: Destination Sequence Distance Routing
  - Use sequence numbers to tackle count-to-infinity
  - Destination node gives an *even* sequence number to its own updates
  - If a neighbour finds a destination down, sends updates with next *odd* sequence number
  - Nodes use routing information with the newest sequence number (or the one with the best metric if the sequence numbers are the same)

Example: DSDV

- Some optimisations
  - Send a “full dump” initially and incremental updates periodically
  - Measure average time between first and best updates for each destination
    - Defer future updates for that time period

Example: Dynamic Source Routing

- On-demand protocol
- Route Discovery
  - Source broadcasts a "route request" message containing the destination and a broadcast ID
  - If an intermediate node does not have a route, it forwards the request, appending its own address
  - Intermediate nodes only forward the first instance of the request they see
  - The destination gets the request with the list of intermediate nodes and sends back this list using the reverse route, or using another route request
  - The source now does source routing using this path
- Route maintenance
  - "Route error" messages for broken links and acknowledgments to ascertain link status

Outline

Attacks

- Routing disruption
  - Loop creation
  - Blackholes (route all packets through self)
  - Blackmail (force blacklisting of a node)
  - Force suboptimal routing
  - Partition the network
  - *Wormholes* (require collusion, hard to tackle)
- Resource consumption
  - Flood control messages

Outline

Introduction

- First set of protocols assume some form of initialisation independent of the ad-hoc network
- Single shared secret
  - One compromised node compromises the network
- Trusted KDC
  - Introduces some infrastructure
  - Single point of failure
- Asymmetric cryptography is an option
  - Expensive for low-capacity nodes
- One-way hash chains

One-way Hash Chains

- Used to authenticate messages from a sender
- We are given a publicly known one-way hash function, H
- Sender generates a random seed, x, and a set of n keys as follows
  - $k_0 = x$
  - $k_i = H(k_{i-1})$
- Receivers are preconfigured with k_n for each sender
- One key per message – sender sends encrypted/signed message and key
- Messages is valid if there if $H^j(key)$ is equal to some previously received key

Outline

TESLA

- Every node has a one-way hash chain
- A node releases keys as per a commonly known schedule
- Requires loose time synchronisation (upto Δ drift)
- Let maximum end-to-end delay be τ
- For each message, sender attaches a keyed MAC using a key that will be not be published before $(\tau + 2\Delta)$ time units from time of sending
- Receiver verifies the TESLA condition
  - The key with which the message has been signed has not yet been published
  - The key will be disclosed soon enough
  - Buffers the packet and waits till the key is published

Ariadne

- Ariadne is based on DSR
- Also assumes pair-wise shared secrets for all source-destination pairs (but can be done without)
- Route request
  - $h_0 = MAC_{SD}(msg)$
  - Source sends $\langle src, dst, id, t_i, h_0, (), () \rangle$
  - An intermediate node, X, verifies that t_i is valid
    - $h_X = H(X, h_{X-1})$
    - $M_X = MAC_{X_{t_i}}(msg)$
    - X sends $\langle src, dst, id, t_i, h_X, (..., X), (..., M_X) \rangle$
  - Receiver can calculate h_0, and can thus validate the request (for the most part)

Ariadne

- Route reply
  - $M_{dst} = MAC_{DS}(msg)$
  - Receiver sends $\langle dst, src, id, t_i, nodelist, hashlist, M_{dst}, () \rangle$
  - Intermediate nodes wait for X_{t_i} to be published and then attach it the list at the end
  - Source can now verify the destination MAC, and that of each node in the route
- Route error
  - If a node finds the next hop is unreachable, sends a Route Error to the source
  - Again use Tesla for authentication
  - $\langle sndr, dst, time, MAC, recentKey \rangle$

Ariadne

- Node in path might not return Route Error messages
  - Get feedback on received packets through some mechanism
  - Use multiple paths, penalising low-reliability paths
  - If an intruder is detected, include a “blacklist” in future route requests
- Route request floods
  - Attacker might flood the network with requests, since these are only finally authenticated by the target
  - Maintain a separate TESLA chain for route requests, and do authentication at neighbours

Outline

Bootstrapping

- Assuming prior initialisation might not be realistic
  - Not all nodes may be administered by a single body
- Hybrid solution
  - Assume at most t nodes can be compromised
  - $(n, t + 1)$ Threshold Cryptography
    - Some nodes have to act as servers
- PGP-like mechanism
- Statistically Unique and Cryptographically Verifiable identifiers

Outline

Bootstrapping using SUCV

- SUCV
  - Every node has a public-private key-pair
  - Address is a hash of the public key
- Again built on DSR
- Route request: source sends $\langle src, dst, id, sig, pubkey, () \rangle$
  - Each intermediate node just appends itself to the list at the end
  - Destination can authenticate the request
- Route reply: destination sends $\langle route, src, dst, id, (a, b, ...), sig, pubkey \rangle$
  - Intermediate nodes cannot tamper, source can verify

Bootstrapping using SUCV

- Route maintenance: intermediate node sends $\langle sndr, dst, sig, pubkey \rangle$
  - Source can verify that the message originated at *sndr*
- This mechanism can be used with SEAD, Ariadne, etc.
- Bugs?
  - Intermediate node can add arbitrary routes during route discovery – maybe each intermediate node can append a signature
  - Need timestamps and loose time-synchronisation to prevent replay attacks

Q&A

Thanks!

References I

- Royer and Toh
  A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks
  IEEE Personal Communications, 1999
- Perkins and Bhagwat
  Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers
  SIGCOMM '94
- Johnson and Maltz
  Dynamic Source Routing in Ad-Hoc Wireless Networks
  Mobile Computing, 1996

References II

- Hu, et. al.
  Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks
  MobiCom '02
- Bobba, et. al
  Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks
  ISR TR 2002