Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks

Yiu-Chun Hu and Adrian Perrig
Carnegie Mellon University
David B. Johnson
Rice University

Introduction

- Ad hoc networks have no fixed structure or base
- In many cases, secure and reliable communication still an important requirement
  - Military networks
  - Disaster relief
  - Mine site operation
- Node mobility and rapid topology changes make routing a difficult problem even when security isn't considered
  - Proactive routing (compensating for topology changes) has high overhead even when the network is static

On-Demand Routing

- Reactive instead of proactive
- Routes discovered only when packets are ready to be sent
  - Generally lower overhead than proactive schemes
  - Can react to topology changes
  - Still maintains efficiency when the network is idle or the topology is static (or relatively static)

Contributions

- Outline attack models on ad hoc network routing and describe new attacks on ad hoc routing
- Design and evaluation of **Ariadne**
  - On-demand secure ad hoc network routing protocol
  - Withstands node compromise
  - Uses only symmetric cryptography

More on Ariadne

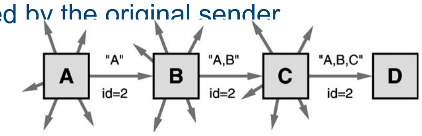
- Authenticated routing using one of three schemes
  - All-pairs shared secret key
  - Shared secret keys combined with broadcast authentication
  - Digital signatures
- TESLA used for broadcast authentication
  - (Does Adrian Perrig have any free time?)
- Based on DSR (Dynamic Source Routing) protocol

Dynamic Source Routing

- Well-studied
- Entirely on demand – routing information only exchanged when a new route is needed
- Two key components
  - Route discovery
  - Route maintenance

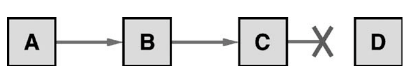
DSR – Route Discovery

- Initiated when a packet is to be sent and no route is in the *route cache* – ROUTE REQUEST
- Nodes either:
  - Rebroadcast packets after appending their address
  - Discard packets if their address already appears or the request is a duplicate
- Target node sends back ROUTE REPLY with list of accumulated addresses
  - Cached by the original sender



DSR – Route Maintenance

- Recall DSR is based on source routing
- If a specified node cannot be reached, a ROUTE ERROR message is sent back to the sender
  - Receipt determined by link-layer acknowledgement, passive acknowledgement, or network-layer acknowledgement
  - Some limited number of retransmissions attempted
- Sender updates its route cache and either uses a different route or initiates another route discovery



Flashback: TESLA

- Broadcast authentication mechanism using only symmetric cryptographic primitives
- Receivers should be able to verify authentication data but not generate it
- Senders and receivers should be loosely time-synchronized
- Senders use one-way key chaining (more on this later)
- Receivers only accept packets generated with secret keys
- Efficient – adds only a single MAC to a message

Network Assumptions

- Physical layer vulnerable, but not considered in this paper
- Links are bidirectional – if node A can receive from node B, it can send as well
- MAC layer attacks exist but are not considered (such as CTS attacks)
- Standard wireless channel

Node Assumptions

- Resource-constrained (Palm PDAs, RIM pagers, etc.)
  - Too constrained for asymmetric cryptography
- When used with TESLA, nodes should be loosely time-synchronized
  - Possibly based on GPS receivers
  - Periodic resynchronization
- Nodes are not tamper-proof

Security Assumptions

- Some mechanism is used to distribute keys regardless of which of the three key schemes is used (pairwise, TESLA, or digital signatures)
  - Key distribution center
  - PKI
  - Preloading
  - Certification authority
- Each node must have an authentic element from route discovery chains (more on this later)
- How is the circular dependency between key setup and routing resolved?

Establishing Authenticated Keys

- Can use a trusted Key Distribution Center (KDC) for key setup between pairs of nodes
- Usually requires some established routing
- Assume nodes share encryption and MAC keys with the KDC
- KDC initiates route discovery with reserved address as the target
  - All nodes return a ROUTE REPLY message
  - Returned routes can be used to send authenticated keys
  - Nodes can request pairwise keys for specific nodes

Attack Models

- Passive
  - Eavesdropping only
  - Not considered (cannot affect routing)
- Active
  - Eavesdropping and false packet injection
  - Can compromise nodes
    - Has cryptographic information for compromised nodes
    - Shares cryptographic information with all owned nodes
  - Notation: Active-*n*-*m*
    - n: Number of compromised nodes
    - m: Number of owned nodes

Attack Models, continued

- Biggest concern is partitioning
- *Active-VC:*
  - Node owns all nodes on a vertex cut through a particular network
  - Requires good nodes to communicate through one or more attacker nodes to reach the “other side”

Types of Attacks

- Routing disruption
  - Routing loops
  - Black holes
  - Gray holes (selective forwarding)
  - Detours (suboptimal routes)
  - Partitions
    - Prevents some nodes from communicating
  - Gratuitous detours
    - Make path through a node appear longer by adding virtual nodes

More Types of Attacks

- False blacklisting
  - In Ariadne, nodes trust only themselves for blacklisting
- Wormhole attacks
- Rushing attacks
  - Disseminates ROUTE REQUEST messages very quickly, increasing probability attacker node is used

- Resource consumption
  - Injecting extra data packets
  - Injecting extra control packets

Ariadne: Design Goals

- Resilience against multiple node compromise
  - Graceful degradation rather than abrupt failure
- Use packet leashes to prevent wormhole and rushing attacks
  - Also works if the nodes are tamper-proof
- Prevent routing disruption attacks by verifying origin and integrity of data
  - Need a suitable authentication mechanism

Authentication

- Needs to be computationally efficient
- Needs to have low network overhead
  - Otherwise, attacks are simple
- Pairwise shared keys
  - Key setup may be expensive
- TESLA broadcast authentication
- Digital signatures
  - For networks with more powerful nodes

Ariadne Route Discovery

- Source S, Destination D
- Target verifies ROUTE REQUEST if a MAC is computed over a timestamp (Key K_{SD})
- Source wants to authenticate each node in the ROUTE REPLY list
  - Also, target wants to authenticate each node in the received ROUTE REQUEST list

Ariadne Route Discovery

- Using TESLA:
  - Each hop authenticates new information
  - Target buffers request until keys are disclosed and includes MAC in ROUTE REPLY
- Using digital signatures:
  - No route discovery chain element needed
  - MAC list becomes signature list
  - No key list required in ROUTE REPLY

Route Discovery using MACs

- Most efficient, but requires pairwise keys
- MAC list based on current node and target
  - Uses pairwise shared key rather than TESLA key
  - Verified at the target
- No key list required in ROUTE REPLY
- Per-hop hashing used to ensure attackers do not remove nodes from list

Route Discovery using TESLA

- All nodes must have shared MAC keys and one authentic TESLA key
- Target can authenticate initiator
- Initiator can authenticate each node in the ROUTE REPLY list
- No intermediate nodes can remove list entries
- Request fields: (ROUTE REQUEST, initiator, target, ID, time interval, hash chain, node list, MAC list)
- Reply fields: (ROUTE REPLY, target, initiator, time interval, node list, MAC list, target MAC, key list)

Route Discovery using TESLA

S:	$h_0 = \mathrm{MAC}_{K_{SD}}(\mathrm{REQUEST}, S, D, id, ti)$
$S \to *$:	$\langle \mathrm{REQUEST}, S, D, id, ti, h_0, (), () \rangle$
A:	$h_1 = H[A, h_0]$
	$M_A = \mathrm{MAC}_{K_{A_{ti}}}(\mathrm{REQUEST}, S, D, id, ti, h_1, (A), ())$
$A \to *$:	$\langle \mathrm{REQUEST}, S, D, id, ti, \underline{\boldsymbol{h_1}}, (\underline{A}), (\underline{\boldsymbol{M_A}}) \rangle$
B:	$h_2 = H[B, h_1]$
	$M_B = \mathrm{MAC}_{K_{B_{ti}}}(\mathrm{REQUEST}, S, D, id, ti, h_2, (A, B), (M_A))$
$B \to *$:	$\langle \mathrm{REQUEST}, S, D, id, ti, \underline{\boldsymbol{h_2}}, (A, \underline{\boldsymbol{B}}), (M_A, \underline{\boldsymbol{M_B}}) \rangle$
C:	$h_3 = H[C, h_2]$
	$M_C = \mathrm{MAC}_{K_{C_{ti}}}(\mathrm{REQUEST}, S, D, id, ti, h_3, (A, B, C), (M_A, M_B))$
$C \to *$:	$\langle \mathrm{REQUEST}, S, D, id, ti, \underline{\boldsymbol{h_3}}, (A, B, \underline{\boldsymbol{C}}), (M_A, M_B, \underline{\boldsymbol{M_C}}) \rangle$
D:	$M_D = \mathrm{MAC}_{K_{DS}}(\mathrm{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C))$
$D \to C$:	$\langle \mathrm{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), \underline{\boldsymbol{M_D}}, () \rangle$
$C \to B$:	$\langle \mathrm{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (\underline{\boldsymbol{K_{C_{ti}}}}) \rangle$
$B \to A$:	$\langle \mathrm{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, \underline{\boldsymbol{K_{B_{ti}}}}) \rangle$
$A \to S$:	$\langle \mathrm{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, K_{B_{ti}}, \underline{\boldsymbol{K_{A_{ti}}}}) \rangle$

Time Intervals

- Time intervals set to pessimistic expected arrival time at the target
- Received time intervals must not be too far in the future and keys must not have been disclosed yet

ROUTE REPLY – Key Disclosure

- Nodes do not forward ROUTE REPLY messages until keys can be disclosed
- Keys are appended to the key list field
- When the initiator receives the ROUTE REPLY message, it verifies:
  - Each key is valid
  - Target MAC is valid
  - Each MAC in the list is valid

Ariadne Route Maintenance

- ROUTE ERROR messages must be authenticated
- Fields: (ROUTE ERROR, sending address, receiving address, time interval, error MAC, recent TESLA key)
  - Sending address detects the error
  - Receiving address is the unreachable node
  - Most recently disclosed TESLA key used

Ariadne Route Maintenace

- Notes that receive ROUTE ERROR messages update their route caches if authentication is successful
- Nodes can only have a finite number of pending ROUTE ERROR messages
- Memory attacks prevented by ensuring the probability information from an ERROR message is in the table is independent of the time that ERROR message was received
  - This attack not valid with digital signature or pairwise key schemes

Misbehaving Nodes

- Detect misbehaving nodes by using a feedback system for packet delivery
  - Best when feedback is sent along the original delivery route of the packet
- When multiple routes are available, use all routes (even known or suspected bad routes) to keep monitoring current
  - Send small fraction of packets along bad routes
- ROUTE REQUEST messages can also include a list of nodes to avoid
  - Adversaries cannot add or remove from this list without being detected

ROUTE REQUEST Flood Attacks

- ROUTE REQUEST not authenticated until it reaches the target
  - Active-1-1 attacker can flood the network
- Need to instantly authenticate ROUTE REQUEST messages
- Use *route discovery chains*
  - Effectively limits the rate of new route discoveries

Route Discovery Chains

- One-way key chains (think TESLA)
- Prevents duplicate ROUTE REQUESTs
  - With high probability all nodes hear all ROUTE REQUESTs
- Another alternative is to schedule the use of route discovery chain elements
  - More computationally intensive, but mitigates attacks even in partitioned networks

Merkle-Winternitz Signatures

- Another alternative for route discovery chains
- Add signature to only one field in the ROUTE REQUEST (target address)
- Adds 20 bytes of overhead per request

Ariadne Optimizations

- TESLA allows for additional caching improvements
  - The MAC relationship is less constrained since broadcast authentication allows for nodes along a path to the target to use that same route
- With symmetric authentication (TESLA or pairwise keys) some fields can be omitted and calculated by the receiver, reducing transmission overhead
  - MAC list

Ariadne Evaluation

- Simulator: ns-2 model with mobility extensions
- Access Control through 802.11 DCF
- TESLA for broadcast authentication
- Pairwise shared keys between nodes
- Simulated with and without overhead optimizations
- Based primarily on DSR model with changes to reflect Ariadne and TESLA parameters
  - Key disclosure intervals
- Also compared with two DSR models
  - Current DSR model
  - Unoptimized DSR model (disabled protocol optimizations not present in Ariadne)

Simulation Parameters

Table 1
Parameters for Ariadne simulations.

Scenario parameters	
Number of nodes	50
Maximum velocity (v_{max})	20 m/s
Dimensions of space	1500 m × 300 m
Nominal radio range	250 m
Source–destination pairs	20
Source data pattern (each)	4 packets/second
Application data payload size	512 bytes/packet
Total application data load	327 kbps
Raw physical link bandwidth	2 Mbps
DSR parameters	
Initial ROUTE REQUEST timeout	2 seconds
Maximum ROUTE REQUEST timeout	40 seconds
Cache size	32 routes
Cache replacement policy	FIFO
TESLA parameters	
TESLA time interval	1 second
Pessimistic end-to-end propagation time (τ)	0.2 seconds
Maximum time synchronization error (Δ)	0.1 seconds
Hash length (ρ)	80 bits

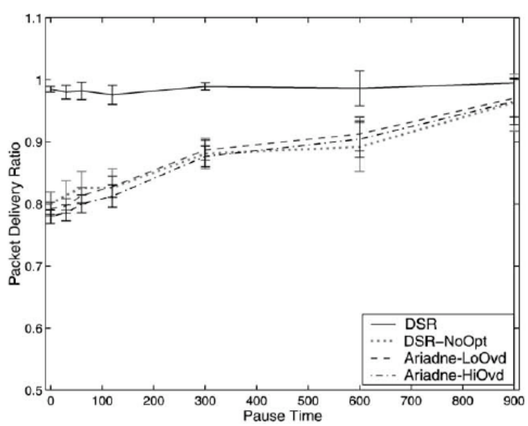
Simulation Parameters

- Nodes use the random waypoint model
  - Fairly standard model
  - Node remains static for a set time, then moves to a randomly determined position with randomly chosen constant velocity
- Note the rectangular simulation space
  - Increases the number of hops used per route

Performance Metrics

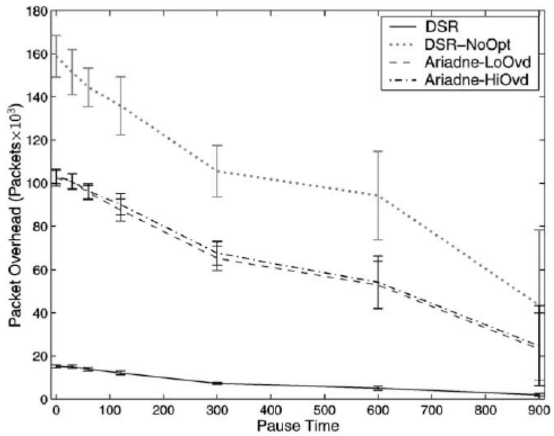
- Packet Delivery Ratio (PDR)
  - Data packets
- Packet Overhead
  - Number of transmissions of routing packets
- Byte Overhead
  - Number of transmissions of overhead bytes
- Mean Packet Latency
- 99.99th Percentile Latency
- Path Optimality

Packet Delivery Ratio



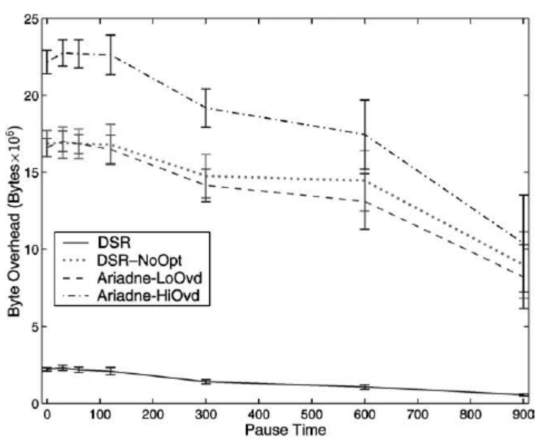
- Noticeable improvement using Ariadne with overhead optimizations
- DSR optimizations also very effective
- Ariadne slightly slower but finds better routes
  - Must be active long enough for ROUTE REPLY
  - Offsets disadvantage from TESLA delay

Packet Overhead



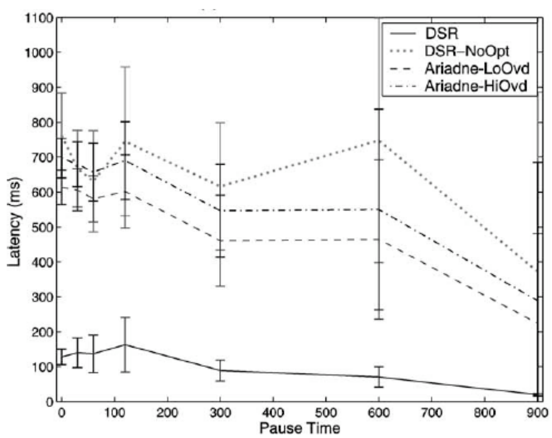
- Ariadne's lower overhead is a result of finding more stable routes
  - Reduces number of ROUTE REQUESTs
  - Reduces number of ROUTE ERRORs
- However, TESLA delay might cause redundant ROUTE ERRORs
  - Not enough to cause major performance hit

Byte Overhead



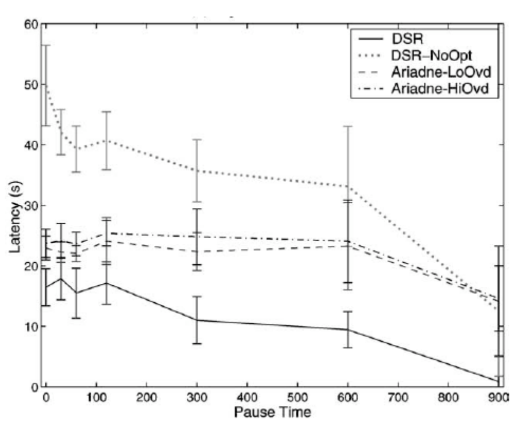
- Ariadne's unoptimized overhead version has significant overhead from additional authentication information
- Optimized Ariadne's overhead offset by its more efficient routing
  - More overhead per packet, but fewer packets sent
  - Outperforms DSR-NoOpt

Mean Packet Latency

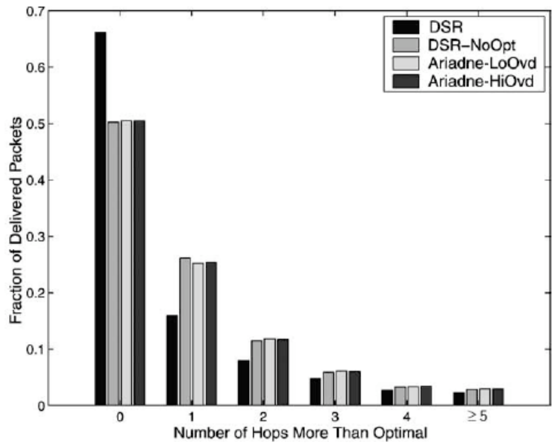


- Latency reduced in Ariadne relative to DSR-NoOpt due to its increased efficiency (less broken links used)
- Significant latency improvement using Ariadne overhead optimizations

99.99th Percentile Latency



Path Optimality



- DSR-NoOpt initiates more route discoveries and hence finds slightly better routes
- On average, Ariadne performance very similar to DSR-NoOpt

Security Analysis

- Minimum broadcast latency path:
  - Path that forwards a discovery the fastest from the source to destination
- Uncompromised route:
  - Path containing only good nodes
- Ariadne will find and use uncompromised routes if they exist
  - At least provided broadcast packets are relatively reliable

Attack Mitigation

- Active-0-x:
  - Replay protection and global MAC keys limit attacks to wormholes and rushing attacks
    - Prevented by packet leashes
- Active-1-1
  - Black/gray holes prevented by per-hop hashing
  - Routing loops prevented by source routing
  - ROUTE REQUEST flooding prevented by authentication and rate limiting through route discovery chains
  - Rushing attacks probabilistically prevented by modifying route discovery

Attack Mitigation

- Active-1-x
  - Wormhole attacks prevented through packet leashes AND GPS (geographic routing)
- Active-y-x
  - False routing (adding other compromised nodes) only works if that is the only route (or the shortest), which is unlikely
  - Forcing multiple route discoveries (forging ROUTE ERROR packets through collusion) not guaranteed to succeed
    - Initiator can include data that must not be altered for the attacker to be part of the path

Attack Mitigation

- Active-VC
  - False floods (holding ROUTE REQUEST messages) defeated by time-synchronizing route discovery chains (requests will be discarded)
  - Black hole (attackers only create routes but drop data packets)

Related Work

- Flooding NPBR
  - Floods all packets through network
  - Authenticates all packets
  - Nodes have allocated bandwidth
  - High overhead
- Security using asymmetric cryptography
  - Wired & wireless applications
  - Subject to verification attacks
- Authenticating link-state updates
- Authenticating routing control packets
- Routing protocol intrusion detection

Conclusion

- Ariadne provides a method for securing on-demand routing in ad hoc networks
  - Uses only symmetric cryptography
  - Resists node compromise
  - Application of security mechanisms is efficient
  - Source routing an efficient mechanism for securing ad hoc networks
  - Provides fine-grained path control

Merits & Contributions

- Ariadne solves a difficult problem in a very efficient manner
  - More efficient than DSR in some cases
  - Achieves very good security properties
- Builds on previous work
  - TESLA
  - DSR

Merits & Contributions

- Results supported by simulations
  - ns-2 is a widely known model
- Compared against a high-performance version (standard DSR) and the most similar modified version for reference
  - Doesn’t pretend to be the best scheme ever designed in every regard
  - Implements an overhead-reducing optimization to Ariadne for additional comparison

Drawbacks

- Implementation is limited to simulation
  - Also, simulation parameters not varied beyond mobility
- Key setup is a difficult problem that is not addressed in sufficient depth
  - Once attackers have been identified, key redistribution to the remaining good nodes could provide some measure of recovery, but this is not addressed