Intruder Identification in Mobile Networks

Bharat Bhargava

CERIAS and CS department

Purdue University

www.cs.purdue.edu/homes/bb

Acknowledgement

- Thanks to all my sponsors in Motorola, Northrup Grumman corporation, Air Force
- Thanks to my students
- Thanks to Infosys Cyber security initiative
- With respect to Bharat Mata whose Matti is my Chandan

and all great people of India.

Intruder Identification in Ad Hoc Networks

- Problem Statement

  Intruder identification in ad hoc networks is the procedure of identifying the user or host that conducts the inappropriate, incorrect, or anomalous activities that threaten the connectivity or reliability of the networks and the authenticity of the data traffic in the networks

Some old Papers with fundamentals:

“*On Security Study of Two Distance Vector Routing Protocols for Mobile Ad Hoc Networks”*, in Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom), 2003.

“*On Vulnerability and Protection of Ad Hoc On-demand Distance Vector Protocol”*, in Proceedings of 10th IEEE International Conference on Telecommunication (ICT), 2003.

Research Motivation

- More than ten routing protocols for Ad Hoc networks have been proposed
    - Incl. AODV, DSR, DSDV, TORA, ZRP
- Research focuses on performance comparison and optimizations such as multicast and multiple path detection
- Research is needed on the security of Ad Hoc networks.
- Applications: Battlefields, disaster recovery.

Research Motivation

- Two kinds of attacks target Ad Hoc network
  - External attacks:
    - MAC Layer jam
    - Traffic analysis
  - Internal attacks:
    - Compromised host sending false routing information
    - Fake authentication and authorization
    - Traffic flooding

Research Motivation

- Protection of Ad Hoc networks
  - Intrusion Prevention
    - Traffic encryption
    - Sending data through multiple paths
    - Authentication and authorization
  - Intrusion Detection
    - Anomaly pattern examination
    - Protocol analysis study

Research Motivation

- Deficiency of intrusion prevention
  - increase the overhead during normal operation period of Ad Hoc networks
  - The restriction on power consumption and computation capability prevent the usage of complex encryption algorithms
  - Flat infrastructure increases the difficulty for the key management and distribution
  - Cannot guard against internal attacks

Research Motivation

- Why intrusion detection itself is not enough
  - Detecting intrusion without isolating the malicious host leaves the protection in a passive mode
  - Identifying the source of the attack may accelerate the detection of other attacks

Research Motivation

- Research problem: Intruder Identification
- Research challenges:
    - How to locate the source of an attack ?
    - How to safely combine the information from multiple hosts and enable individual host to make decision by itself ?
    - How to achieve consistency among the conclusions of a group of hosts ?

Related Work

- Vulnerability model of ad hoc routing protocols [Yang *et al.*, SASN ’03]
- A generic multi layer integrated IDS structure [Zhang and Lee, MobiCom ’00]
- IDS combining with trust [Albert *et al.*, ICEIS ’02]
- Information theoretic measures using entropy [Okazaki *et al*., SAINT ’02]
- SAODV adopts both hash chain and digital signature to protect routing information [Zapata *et al*, WiSe’03]
- Security-aware ad hoc routing [Kravets *et al*, MobiHOC’01]

Related Work in wired Networks

- Secure routing / intrusion detection in wired networks
  - Routers have more bandwidth and CPU power
  - Steady network topology enables the use of static routing and default routers
  - Large storage and history of operations enable the system to collect enough information to extract traffic patterns
  - Easier to establish trust relation in the hierarchical infrastructure

Related Work in wired Networks

- Attack on RIP (Distance Vector)
  - False distance vector
- Solution (Bellovin 89)
  - Static routing
  - Listen to specific IP address
  - Default router
  - Cannot apply in Ad Hoc networks

Related Work in wired Networks

- Attack on OSPF (Link State)
  - False connectivity
  - Attack on Sequence Number
  - Attack on lifetime
- Solution
  - JiNAO:NCSU and MCNC
  - Encryption and digital signature

Related Work in Ad Hoc Networks

- Lee at GaTech summarizes the difficulties in building IDS in Ad Hoc networks and raises questions:
  - what is a good architecture and response system?
  - what are the appropriated audit data sources?
  - what is the good model to separate normal and anomaly patterns?
- Haas at Cornell lists the 2 challenges in securing Ad Hoc networks:
  - secure routing
  - key management service

Related Work in Ad Hoc Networks

- Agrawal at University of Cincinnati presents the general security schemes for the secure routing in Ad Hoc networks
- Nikander at Helsinki discusses the authentication, authorization, and accounting in Ad Hoc networks
- Bhargavan at UIUC presents the method to enhance security by dynamic virtual infrastructure
- Vaidya at UIUC presents the idea of securing Ad Hoc networks with directional antennas

Related Work ongoing projects

- TIARA: Techniques for Intrusion Resistant Ad-Hoc Routing Algorithm (DARPA)
    - develop general design techniques
    - focus on DoS attack
    - sustain continued network operations
- Secure Communication for Ad Hoc Networking (NSF)
    - Two main principles:
        - redundancy in networking topology, route discovery and maintenance
        - distribution of trust, quorum for trust

Related Work ongoing projects

- On Robust and Secure Mobile Ad Hoc and Sensor Network (NSF)
    - local route repair
    - performance analysis
    - malicious traffic profile extraction
    - distributed IDs
    - proposed a scalable routing protocol
- Adaptive Intrusion Detection System (NSF)
    - enable data mining approach
    - proactive intrusion detection
    - establish algorithms for auditing data

Evaluation Criteria

- Accuracy
    - False coverage: Number of normal hosts that are incorrectly marked as suspected.
    - False exclusion: Number of malicious hosts that are not identified as such.
- Overhead
    - Overhead measures the increases in control packets and computation costs for identifying the attackers (e.g. verifying signed packets, updating blacklists).
    - Workload of identifying the malicious hosts in multiple rounds

Evaluation Criteria - cont.

- Effectiveness
    - Effectiveness: Increase in the performance of ad hoc networks after the malicious hosts are identified and isolated. Metrics include the increase of the packet delivery ratio, the decrease of average delay, or the decrease of normalized protocol overhead (control packets/delivered packets).
- Robustness
    - Robustness of the algorithm: Its ability to resist different kinds of attacks.

Assumptions

A1. Every host can be uniquely identified, and its ID cannot be changed throughout the lifetime of the ad hoc network. The ID is used in the identification procedure.

A2. A malicious host has total control on the time, the target and the mechanism of an attack. The malicious hosts continue attacking the network.

A3. Digital signature and verification keys of the hosts have been distributed to every host. The key distribution in ad hoc networks is a tough problem and deserves further research. Several solutions have been proposed. We assume that the distribution procedure is finished, so that all hosts can examine the genuineness of the signed packets.

A4. Every host has a local blacklist to record the hosts it suspects. The host has total control on adding and deleting elements from its list. For the clarity of the remainder of this paper, we call the real attacker as “malicious host”, while the hosts in blacklists are called “suspected hosts”.

Applying Reverse Labeling Restriction to Protect AODV

- Introduction to AODV
- Attacks on AODV and their impacts
- Detecting False Destination Sequence Attack
- Reverse Labeling Restriction Protocol
- Simulation results

Introduction to AODV

- Introduced in 97 by Perkins at NOKIA, Royer at UCSB
- 12 versions of IETF draft in 4 years, 4 academic implementations, 2 simulations
- Combines on-demand and distance vector
- Broadcast Route Query, Unicast Route Reply
- Quick adaptation to dynamic link condition and scalability to large scale network
- Support multicast

Ideas

- Monitor the sequence numbers in the route request packets to detect abnormal conditions
- Apply reverse labeling restriction to identify and isolate attackers
- Combine local decisions with knowledge from other hosts to achieve consistent conclusions
- Combine with trust assessment methods to improve robustness

Security Considerations for AODV

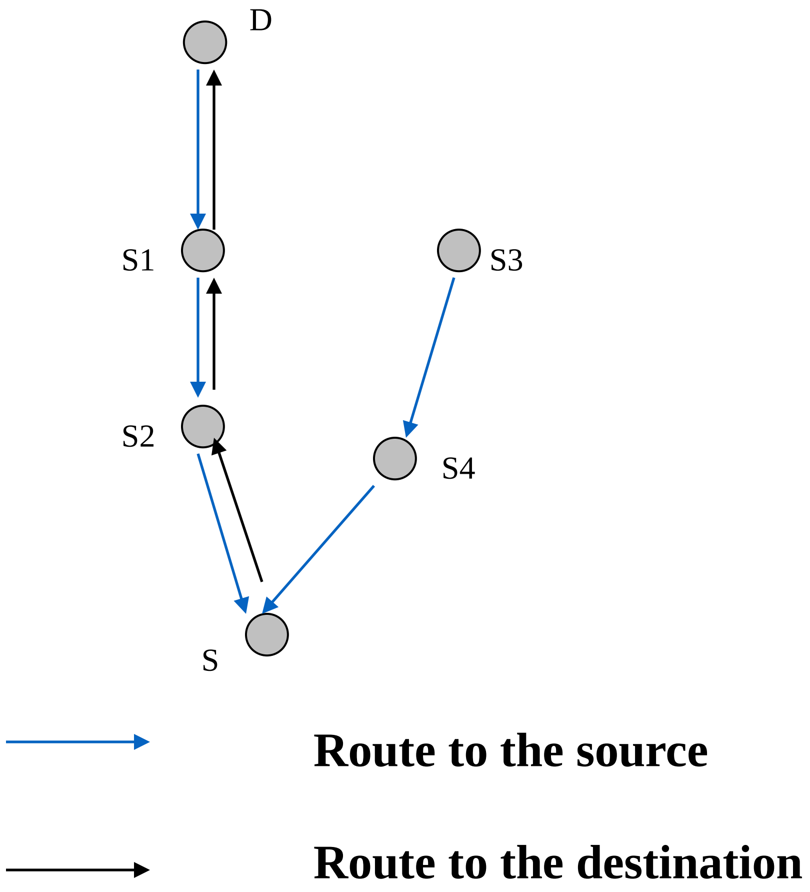
"AODV does not specify any special security measures. Route protocols, however, are prime targets for impersonation attacks. If there is danger of such attacks, AODV control messages must be protected by use of authentication techniques, such as those involving generation of unforgeable and cryptographically strong message digests or digital signatures. "

- http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-11.txt

Message Types in AODV

- RREQ: route request
- RREP: route reply
- RERR: route error

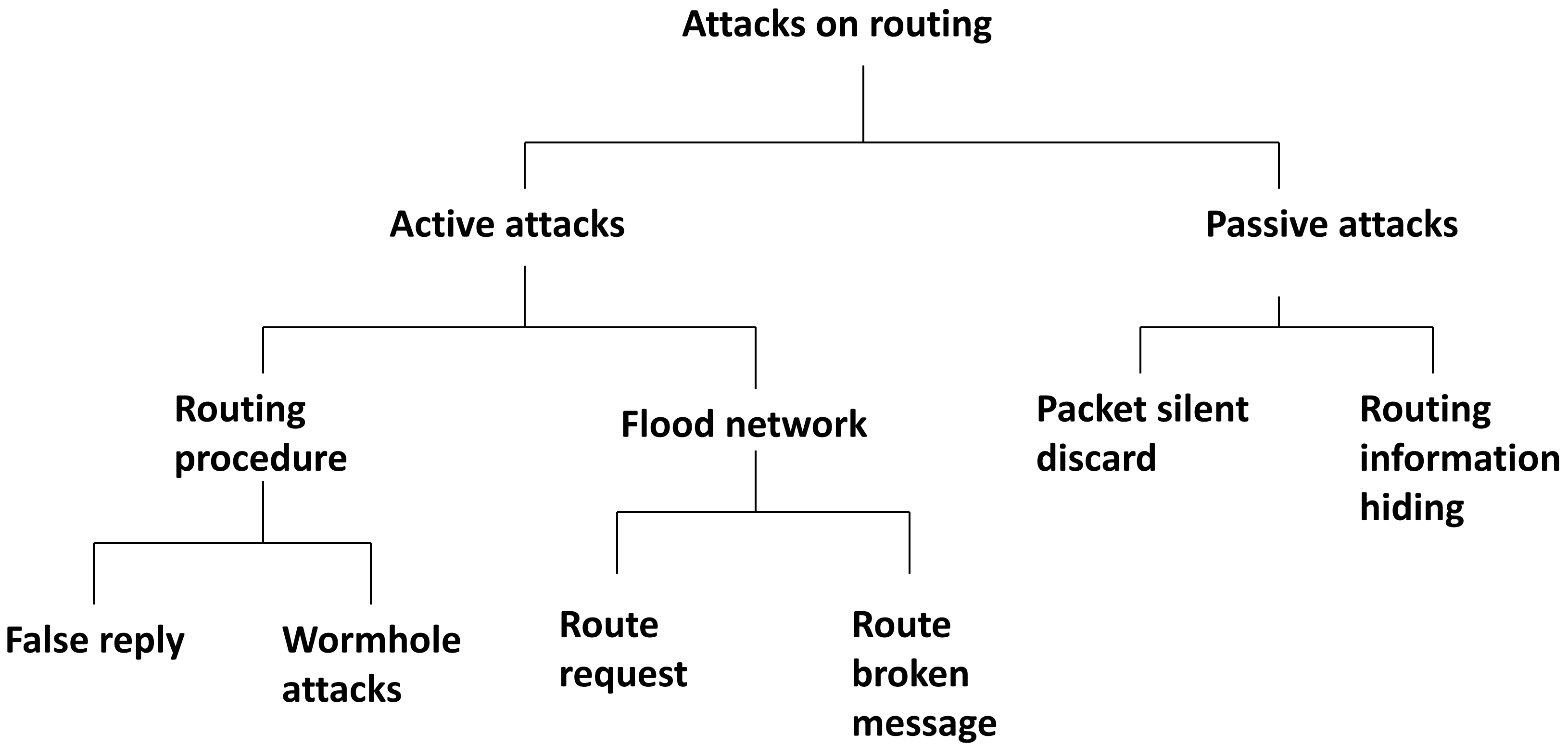
Route Discovery in AODV (An Example)



Route to the source

Route to the destination

Attacks on routing in mobile ad hoc networks



Attacks on routing

Active attacks

Passive attacks

Routing procedure

Flood network

Packet silent discard

Routing information hiding

False reply

Wormhole attacks

Route request

Route broken message

Attacks on AODV

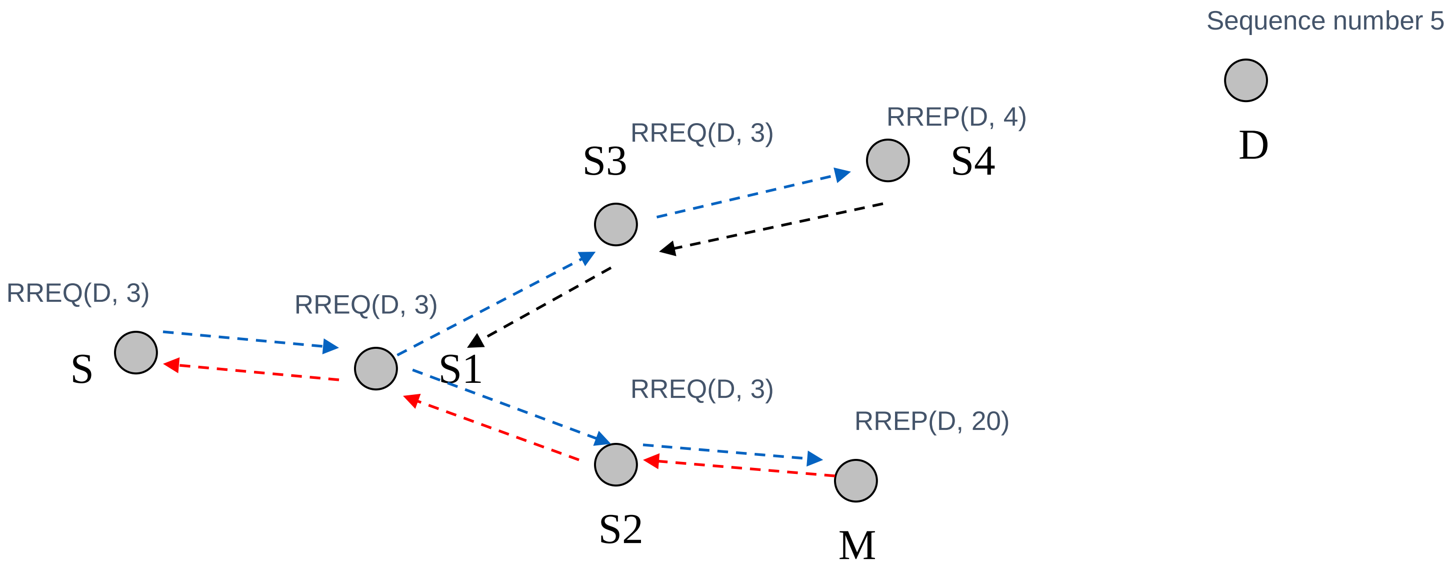
- Route request flooding
  - query non-existing host (RREQ will flood throughout the network)
- False distance vector
  - reply “one hop to destination” to every request and select a large enough sequence number
- False destination sequence number
  - select a large number (even beat the reply from the real destination)
- Wormhole attacks
  - tunnel route request through wormhole and attract the data traffic to the wormhole
- Coordinated attacks
  - The malicious hosts establish trust to frame other hosts, or conduct attacks alternatively to avoid being identified

Impacts of Attacks on AODV

- We simulate the attacks and measure their impacts on packet delivery ratios and protocol overhead

	Packet Delivery Ratio	Control packet / data packet
No Attacks	96%	0.38
Vicious Flooding	91%	2.93
False Distance	75%	0.38
False Destination Sequence	53%	0.66
Wormhole	61%	0.41

False Destination Sequence Attack

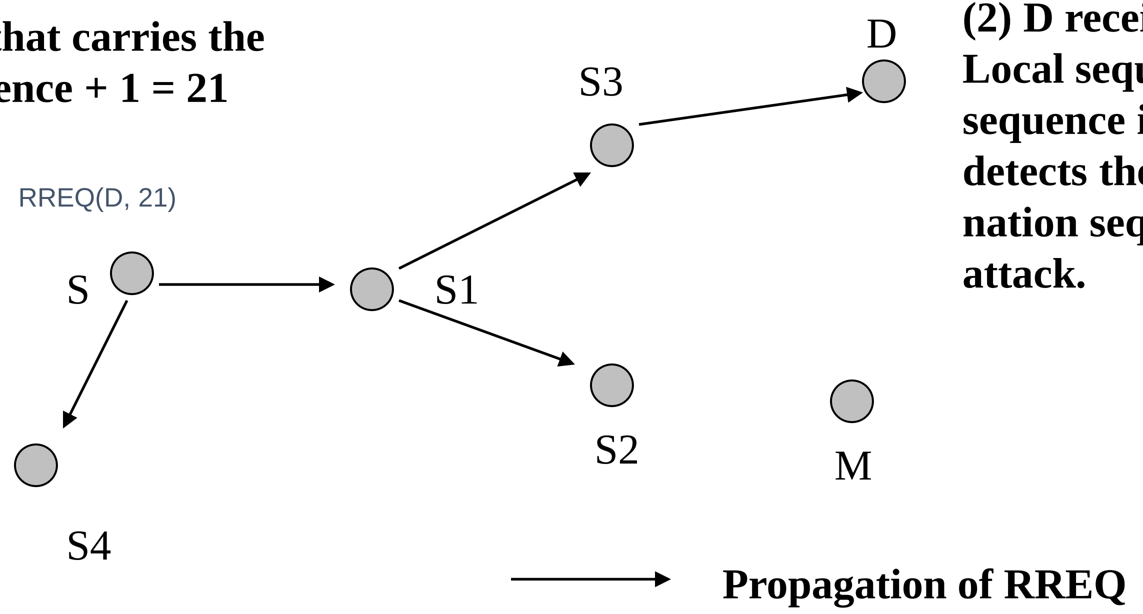


Packets from S to D are sinking at M.

- During Route Rediscovery, False Destination Sequence Number Attack Is Detected, S needs to find D again.

Node movement breaks the path from S to M (trigger route rediscovery).

(1). S broadcasts a request that carries the old sequence + 1 = 21

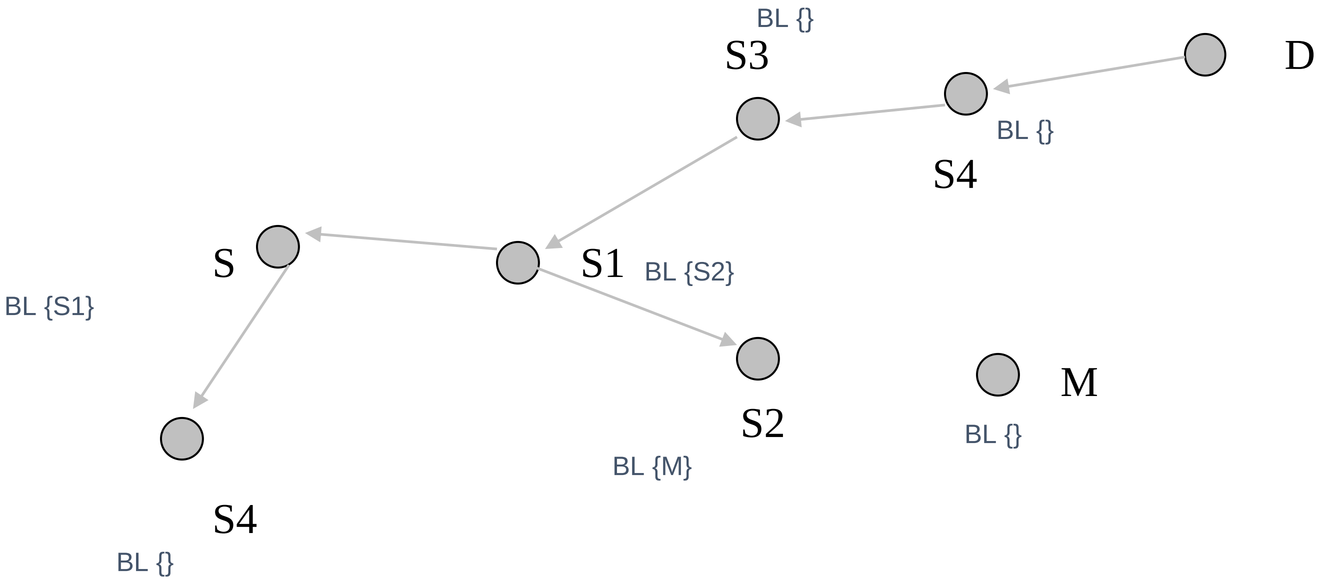


(2) D receives the RREQ. Local sequence is 5, but the sequence in RREQ is 21. D detects the false destination sequence number attack.

Reverse Labeling Restriction (RLR)

Blacklists are updated after an attack is detected.

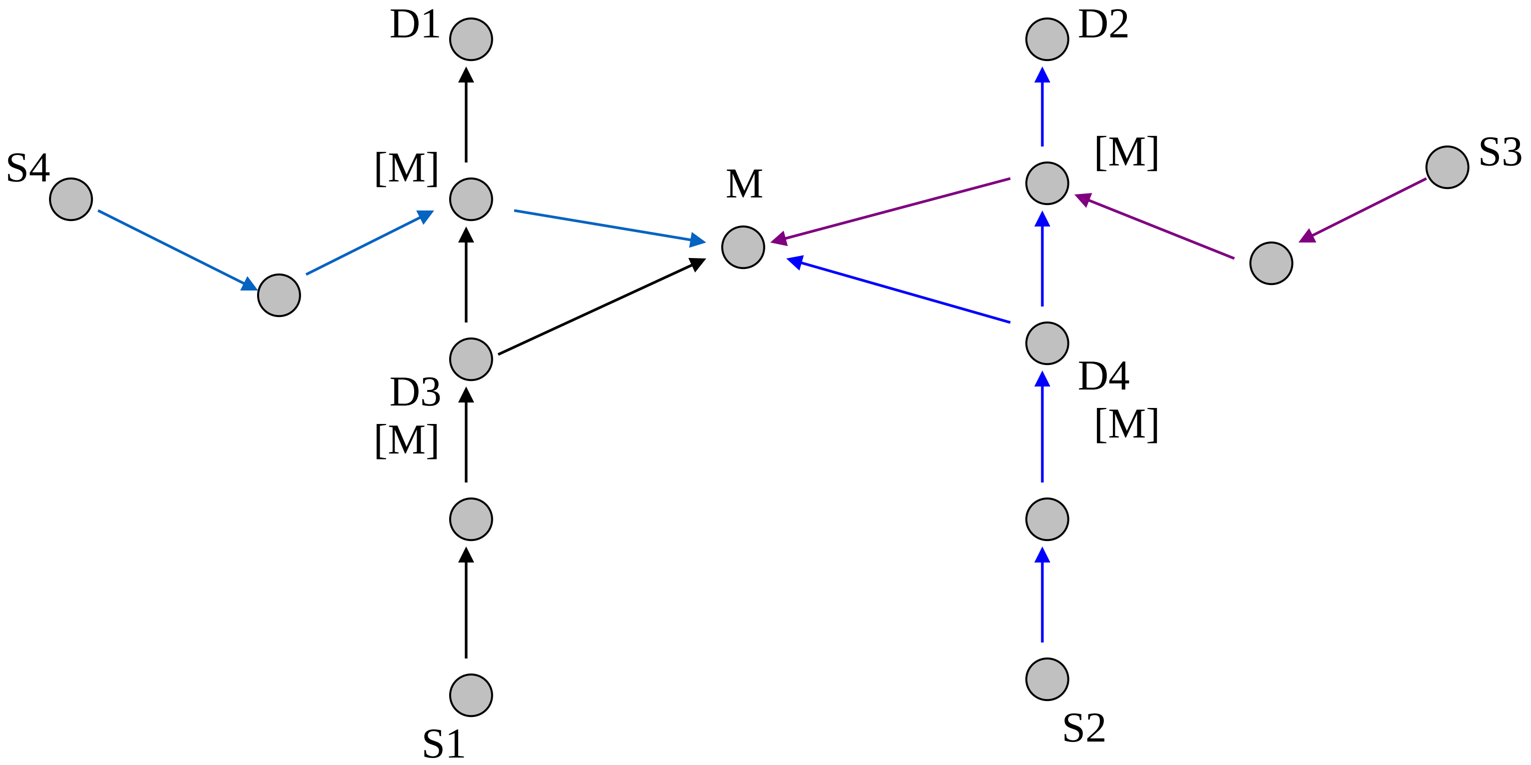
- Basic Ideas
  - Every host maintains a blacklist to record suspicious hosts who gave wrong route related information.
  - The destination host will broadcast an INVALID packet with its signature. The packet carries the host’s identification, current sequence, new sequence, and its own blacklist.
  - Every host receiving this packet will examine its route entry to the destination host. The previous host that provides the false route will be added into this host’s blacklist.



INVALID (D, 5, 21,
BL{}, Signature)

Correct destination sequence number is broadcasted.

Blacklist at each host in the path is determined.



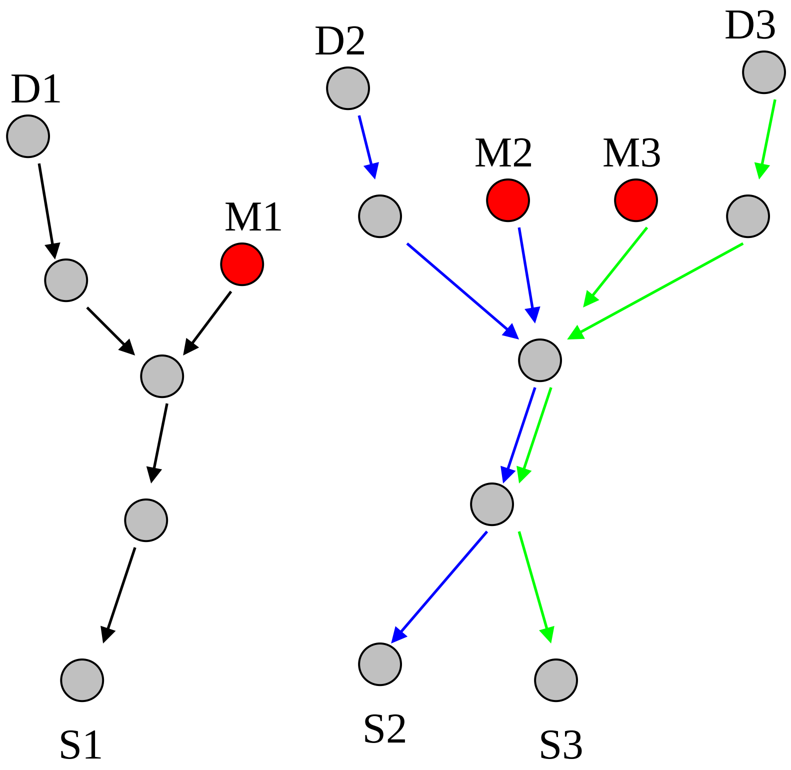
M attacks 4 routes (S1-D1, S2-D2, S3-D3, and S4-D4). When the first two false routes are detected, D3 and D4 add M into their blacklists. When later D3 and D4 become victim destinations, they will broadcast their blacklists, and every host will get two votes that M is malicious host.

Malicious site is in blacklists of multiple destination hosts.

Combine Local Decisions with Knowledge from Other Hosts

- When a host is destination of a route and is victim by any malicious host, it will broadcast its blacklist.
- Each host obtains blacklists from victim hosts.
- If M is in multiple blacklists, M is classified as a malicious host based on certain threshold.
- Intruder is identified.
- Trust values can be assigned to other hosts based on past information.

Acceleration in Intruder Identification



Coordinated attacks by M1, M2, and M3

Multiple attackers trigger more blacklists to be broadcasted by D1, D2, D3.

Reverse Labeling Restriction (RLR)

- Update Blacklist by Broadcasted Packets from Destinations under Attack
  - Next hop on the false route will be put into local blacklist, and a counter increases. The time duration that the host stays in blacklist increases exponentially to the counter value.
  - When timer expires, the suspicious host will be released from the blacklist and routing information from it will be accepted.

Deal With Hosts in Blacklist

- Packets from hosts in blacklist
    - Route request: If the request is from suspicious hosts, ignore it.
    - Route reply: If the previous hop is suspicious and the query destination is not the previous hop, the reply will be ignored.
    - Route error: Will be processed as usual. RERR will activate re-discovery, which will help to detect attacks on destination sequence.
    - Broadcast of INVALID packet: If the sender is suspicious, the packet will be processed but the blacklist will be ignored.

Attacks of Malicious Hosts on RLR

- Attack 1: Malicious host M sends false INVALID packet
    - Because the INVALID packets are signed, it cannot send the packets in other hosts' name
    - If M sends INVALID in its own name
        - If the reported sequence number is greater than the real sequence number, every host ignores this attack
        - If the reported sequence number is less than the real sequence number, RLR will converge at the malicious host. M is included in blacklist of more hosts. M accelerated the intruder identification directing towards M.

- Attack 2: Malicious host M frames other innocent hosts by sending false blacklist
  - If the malicious host has been identified, the blacklist will be ignored
  - If the malicious host has not been identified, this operation can only make the threshold lower. If the threshold is selected properly, it will not impact the identification results.
  - Combining trust can further limit the impact of this attack.

- Attack 3: Malicious host M only sends false destination sequence about some special host
  - The special host will detect the attack and send INVALID packets.
  - Other hosts can establish new routes to the destination by receiving the INVALID packets.

Experimental Studies of RLR

- The experiments are conducted using ns2.
- Various network scenarios are formed by varying the number of independent attackers, number of connections, and host mobility.
- The examined parameters include:
  - Packet delivery ratio
  - Identification accuracy: false positive and false negative ratio
  - Communication and computation overhead

Simulation Parameter

Simulation duration	1000 seconds
Simulation area	1000 * 1000 m
Number of mobile hosts	30
Transmission range	250 m
Pause time between the host reaches current target and moves to next target	0 – 60 seconds
Maximum speed	5 m/s
Number of CBR connection	25/50
Packet rate	2 pkt / sec

Experiment 1: Measure the Changes in Packet Delivery Ratio

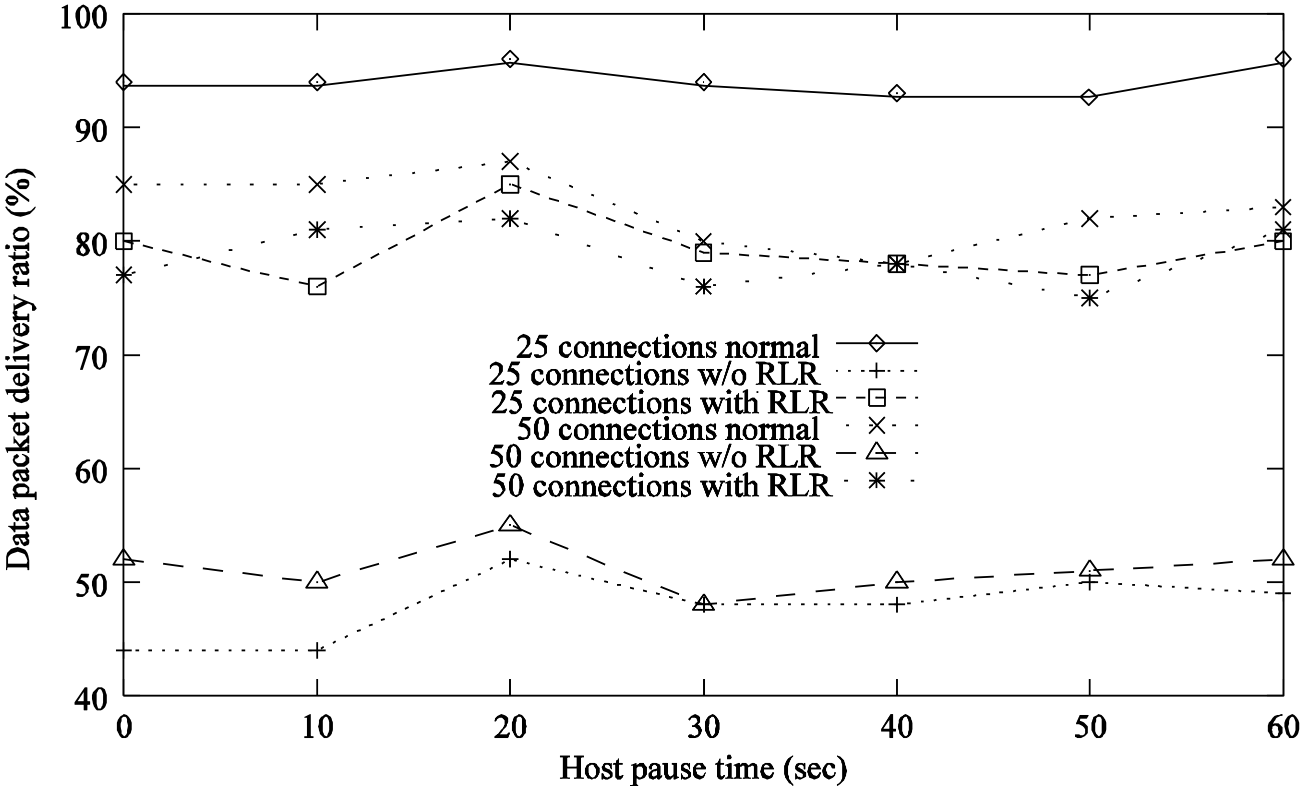
Purpose: investigate the impacts of host mobility, number of attackers, and number of connections on the performance improvement brought by RLR

Input parameters: host pause time, number of independent attackers, number of connections

Output parameters: packet delivery ratio

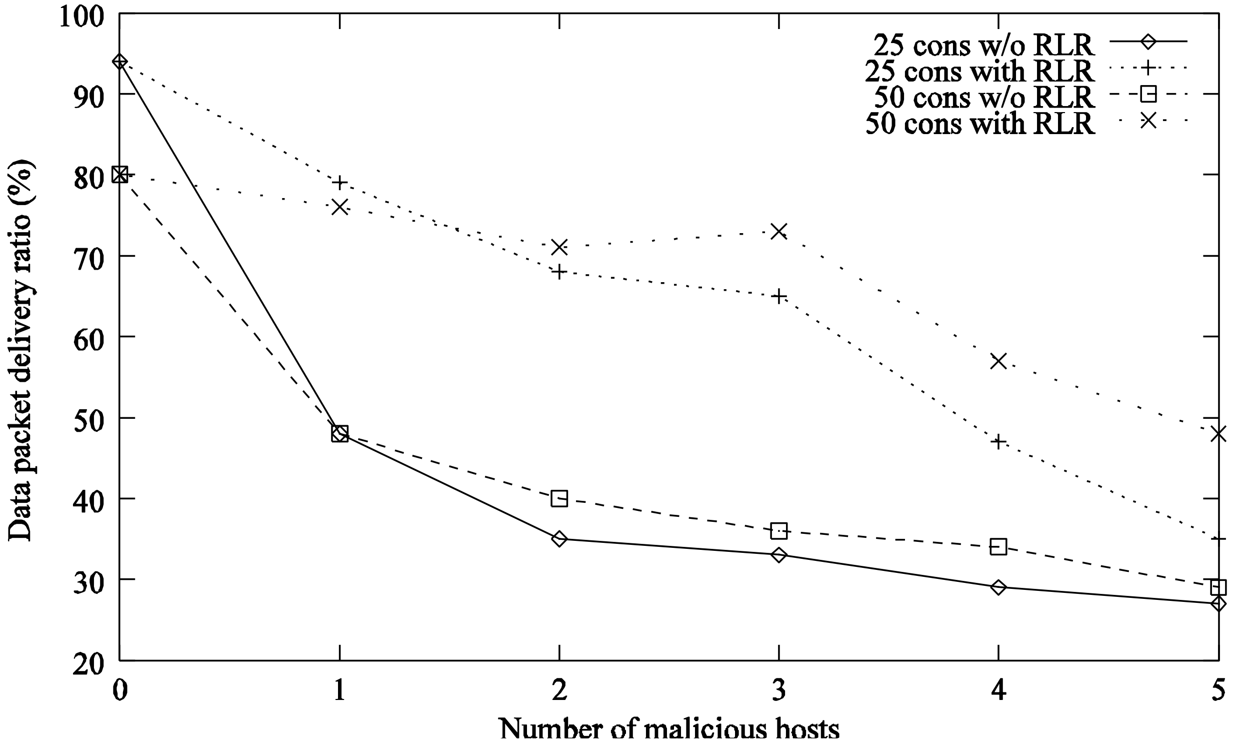
Observation: When only one attacker exists in the network, RLR brings a 30% increase in the packet delivery ratio. When multiple attacker exist in the system, the delivery ratio will not recover before all attackers are identified.

Increase in Packet Delivery Ratio: Single Attacker



X-axis is host pause time, which evaluates the mobility of host. Y-axis is delivery ratio. 25 connections and 50 connections are considered. RLR brings a 30% increase in delivery ratio. 100% delivery is difficult to achieve due to network partition, route discovery delay and buffer.

Increase in Packet Delivery Ratio: Multiple Attackers



X-axis is number of attackers. Y-axis is delivery ratio. 25 connections and 50 connections are considered. RLR brings a 20% to 30% increase in delivery ratio.

Experiment 2: Measure the Accuracy of Intruder Identification

Purpose: investigate the impacts of host mobility,   number of attackers ,and connection scenarios on the detection accuracy of RLR

Input parameters: number of independent attackers, number of connections, host pause time

Output parameters: false positive alarm ratio, false negative alarm ratio

Observation: The increase in connections may improve the detection accuracy of RLR. When multiple attackers exist in the network, RLR has a high false positive ratio.

Accuracy of RLR: Single Attacker

	30 hosts, 25 connections		30 hosts, 50 connections	
Host Pause time (sec)	# of normal hosts identify the attacker	# of normal hosts marked as malicious	# of normal hosts identify the attacker	# of normal hosts marked as malicious
0	24	0.22	29	2.2
10	25	0	29	1.4
20	24	0	25	1.1
30	28	0	29	1.1
40	24	0	29	0.6
50	24	0.07	29	1.1
60	24	0.07	24	1.0

The accuracy of RLR when there is only one attacker in the system

Accuracy of RLR: Multiple Attackers

	30 hosts, 25 connections		30 hosts, 50 connections	
# of attackers	# of normal hosts identify all attackers	# of normal hosts marked as malicious	# of normal hosts identify all attackers	# of normal hosts marked as malicious
1	28	0	29	1.1
2	28	0.65	28	2.6
3	25	1	27	1.4
4	21	0.62	25	2.2
5	15	0.67	19	4.1

The accuracy of RLR when there are multiple attackers

Experiment 3: Measure the Communication Overhead

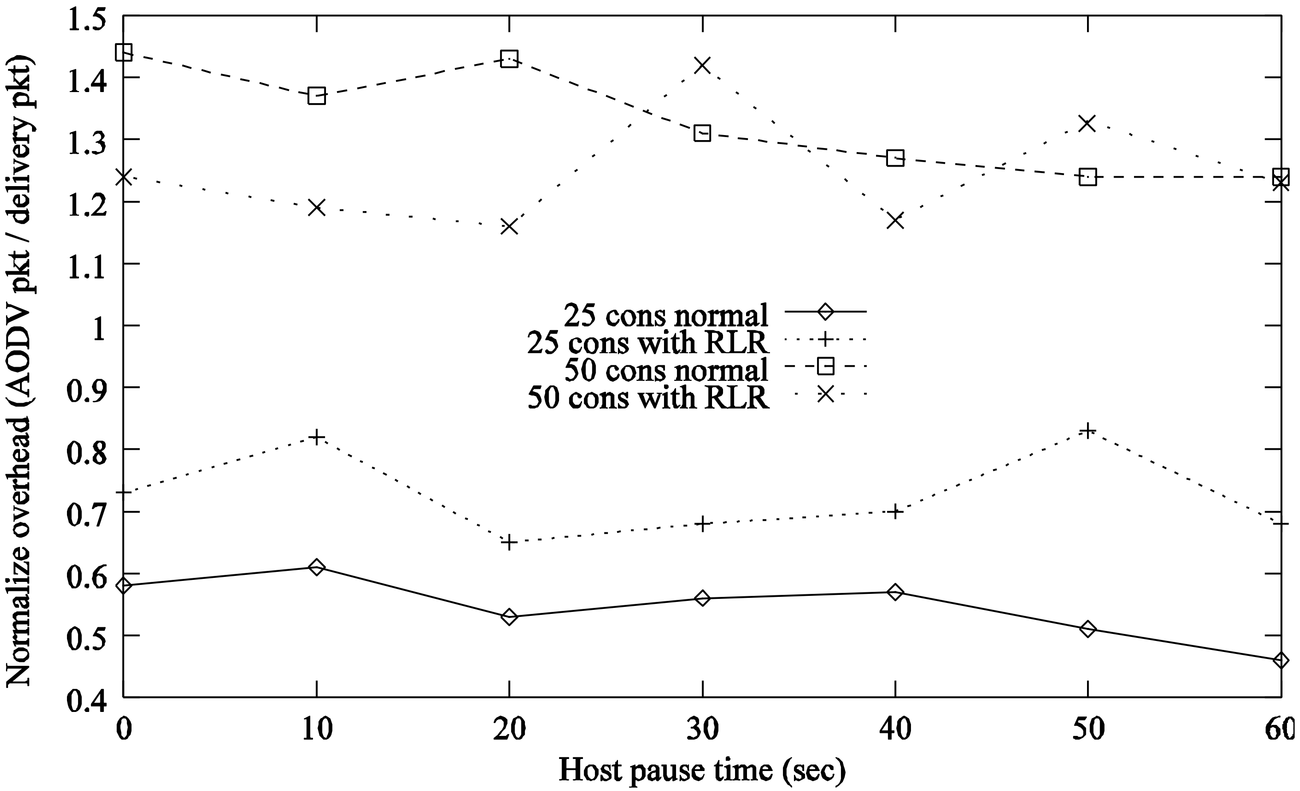
Purpose: investigate the impacts of host mobility and connection scenarios on the overhead of RLR

Input parameters: number of connections, host pause time

Output parameters: control packet overhead

Observation: When no false destination sequence attacks exist in the network, RLR introduces small packet overhead into the system.

Control Packet Overhead



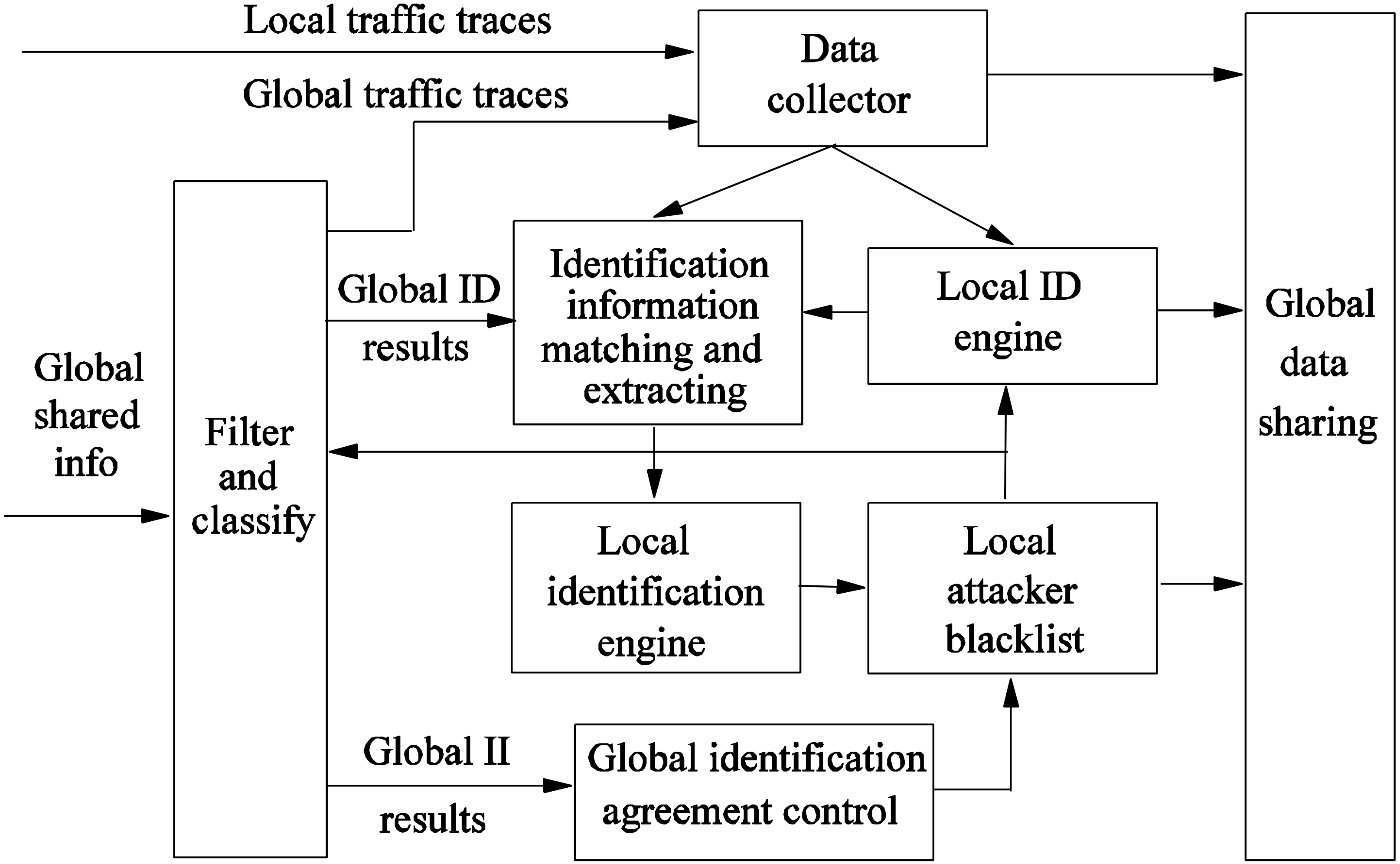
X-axis is host pause time, which evaluates the mobility of host. Y-axis is normalized overhead (# of control packet / # of delivered data packet). 25 connections and 50 connections are considered. RLR increases the overhead slightly.

Research Opportunities: Improve Robustness of RLR

- Protect the good hosts from being framed by malicious hosts
    - The malicious hosts can frame the good hosts by putting them into blacklist.
    - By lowering the trust values of both complainer and complainee, we can restrict the impacts of the gossip distributed by the attackers.
- Avoid putting every host into blacklist
    - Combining the host density and movement model, we can estimate the time ratio that two hosts are neighbors
    - The counter for a suspicious host decreases as time passes
    - Adjusting the decreasing ratio to control the average percentage of time that a host stays in the blacklist of another host

- Defend against coordinated attacks
    - The behaviors of collusive attackers show Byzantine manners. The malicious hosts may establish trust to frame other hosts or conduct attacks alternatively to avoid being identified.
    - Look for the effective methods to defend against such attacks. Possible research directions include:
        - Apply classification methods to detect the hosts that have similar behavior patterns
        - Study the behavior histories of the hosts that belong to the same group and detect the pattern of malicious behavior (time-based, order-based)

An Architecture of Intruder Identification Agent



- Intruder identification can be applied to detect more attacks in ad hoc networks:
  - DoS attacks
  - Malicious discard
  - Trust abuse and privacy violation
- Reverse labeling mechanism can be applied to identify the attackers that
  - Disseminate false routing information
  - Discard data packets
  - Generate gossip to destroy other hosts’ reputation

Conclusions on Intruder Identification

- False destination sequence attacks can be detected by the anomaly patterns of the sequence numbers
- Reverse labeling method can reconstruct the false routing tree
- Isolating the attackers brings a sharp increase in network performance
- On going research will improve the robustness of the mechanism and the accuracy of identification

Trusted Router and Protection Against Collaborative Attacks

- Characterizing collaborative/coordinated attacks
- Types of collaborative attacks
- Identifying Malicious activity
- Identifying Collaborative Attack

Collaborative Attacks

Informal definition:

"Collaborative attacks (CA) occur when more than one attacker or running process synchronize their actions to disturb a target network"

Collaborative Attacks (cont’d)

- Forms of collaborative attacks
  - Multiple attacks occur when a system is disturbed by more than one attacker
  - Attacks in quick sequences is another way to perpetrate CA by launching sequential disruptions in short intervals
  - Attacks may concentrate on a group of nodes or spread to different group of nodes just for confusing the detection/prevention system in place
  - Attacks may be long-lived or short-lived
  - Collaborative attacks can be launched intentionally or accidentally
  - Attacks on routing

Collaborative Attacks (cont’d)

- Open issues
  - Comprehensive understanding of the coordination among attacks and/or the collaboration among various attackers
  - Characterization and Modeling of CAs
  - Intrusion Detection Systems (IDS) capable of correlating CAs
  - Coordinated prevention/defense mechanisms

Collaborative Attacks (cont’d)

- From a low-level technical point of view, attacks can be categorized into:
  - Attacks that may overshadow (cover) each other
  - Attacks that may diminish the effects of others
  - Attacks that interfere with each other
  - Attacks that may expose other attacks
  - Attacks that may be launched in sequence
  - Attacks that may target different areas of the network
  - Attacks that are just below the threshold of detection but persist in large numbers

Examples of Attacks that can Collaborate

- Denial-of-Messages (DoM) attacks
- Blackhole attacks
- Wormhole attacks
- Replication attacks
- Sybil attacks
- Rushing attacks
- Malicious flooding

We are investigating the interactions among these forms of attacks

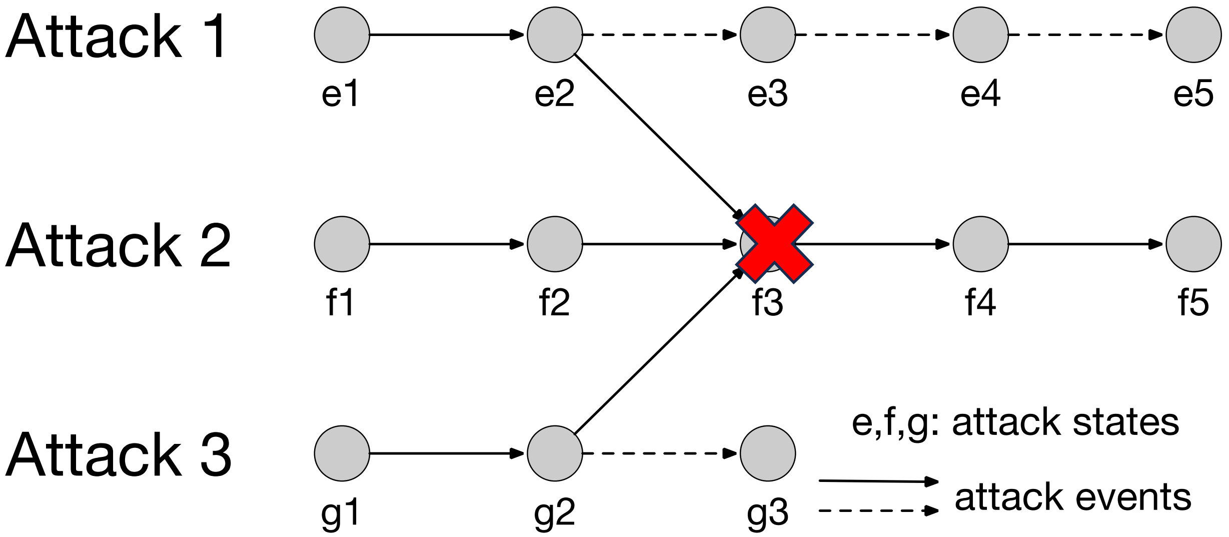
Example of probably **incompatible** attacks:

Wormhole attacks need fast connections, but **DoM** attacks reduce bandwidth!

Current Proposed Solutions

- Blackhole attack detection
  - Reverse Labeling Restriction (RLR)
- Wormhole Attacks: defense mechanism
  - E2E detector and Cell-based Open Tunnel Avoidance (COTA)
- Sybil Attack detection
  - Light-weight method based on hierarchical architecture
- Modeling Collaborative Attacks using Causal Model
- Detecting Collaboration using Machine Learning

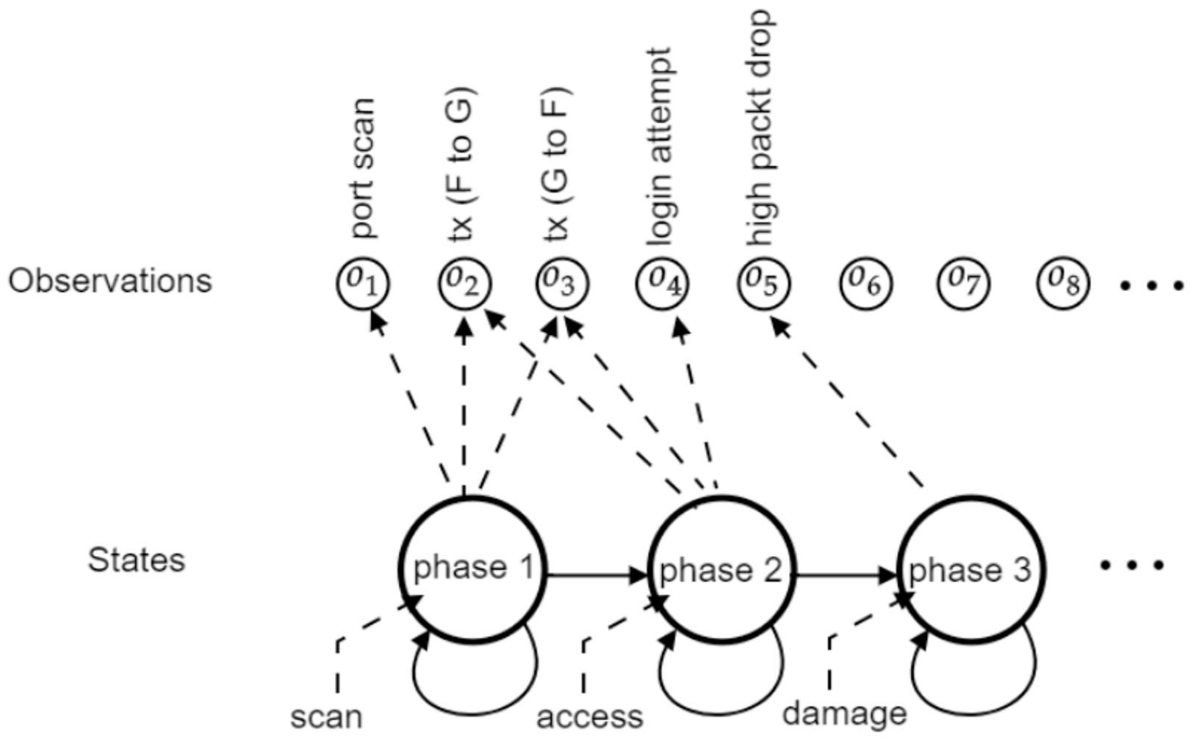
Modeling collaborative attacks with causal graph model



A collaborative attack can be modeled as a causal graph model <*S*, *E*, *M*, *L*>, where

- *S* is the set of attack states
- *E* is the set of events triggering state transition, which can be further defined by
  - Message exchange between attackers where messages are from set *E*, and
  - Local attack operations from operation set *L*
- The goal is to identify the malicious event sequences of collaborative attacks and stop it from reaching the key state (f3 in the above example)

Detecting Collaboration using Hidden Markov Model

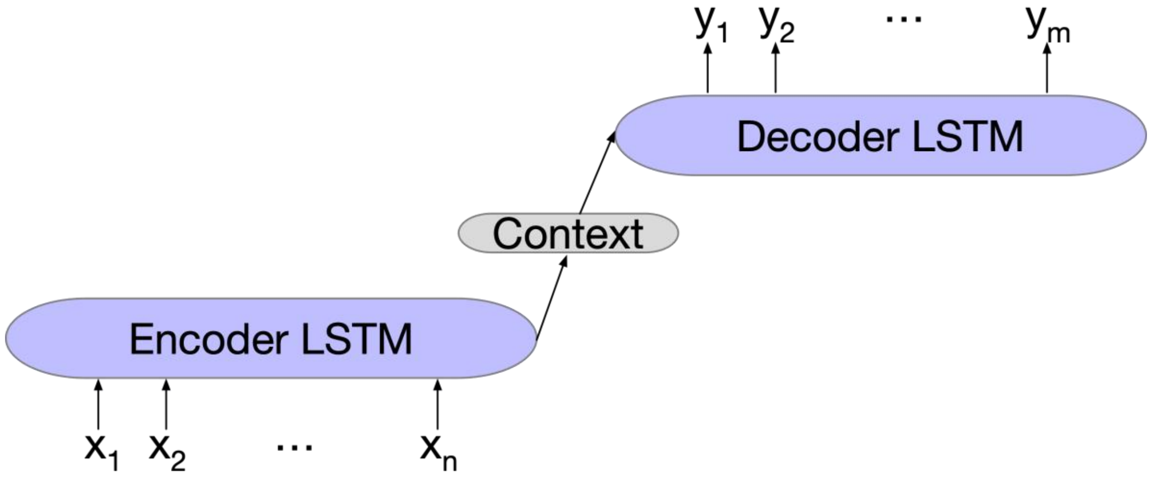


- **Monitors event patterns to identify collaborative attacks.**
- Processes network events as key inputs for analysis.
- Predicts attacks (hidden state) from observations
- Relies on current observation and a fixed number of previous hidden states.
- Effective even with limited data
- Example: A naive collaborative attack can be caught by monitoring a simple sequence of events
  - port scan by one node, messages exchanged between two nodes, login attempt by another node.

Detecting Collaboration using Long Short-Term Memory

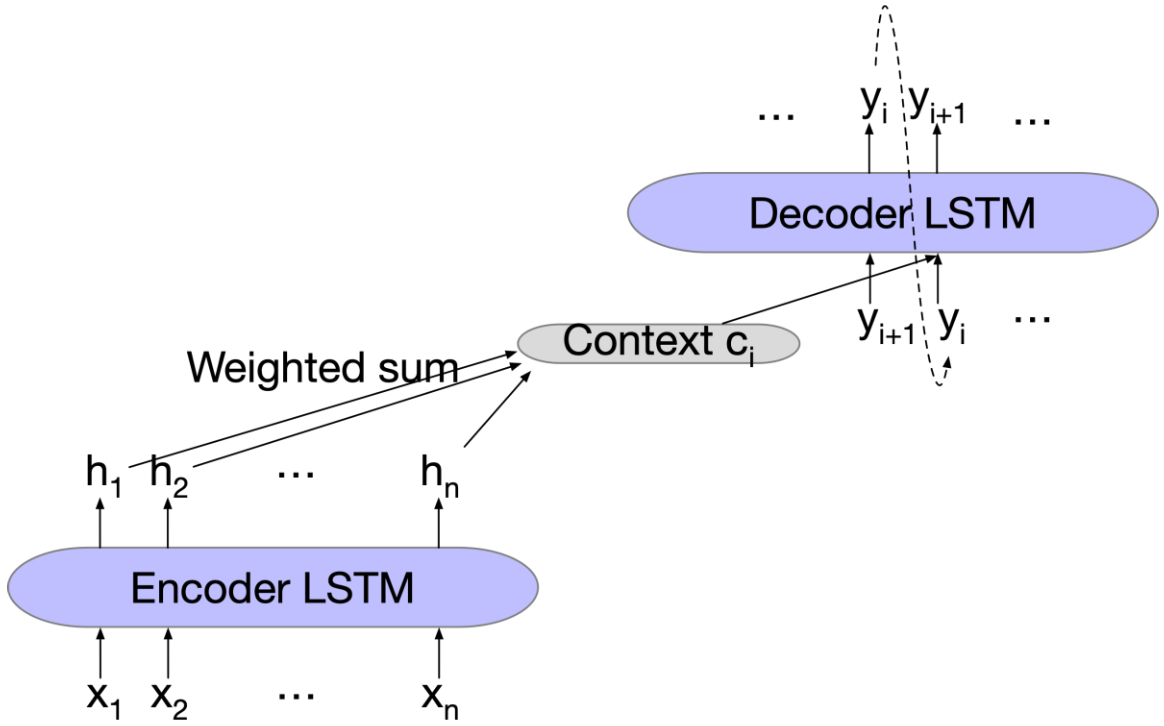
- Instead of leveraging a fixed number of previous hidden states as HMM, an LSTM model uses all previous states and weights them by their importance for prediction.
- Attention-based LSTM will be used
- Compared with HMM, LSTM needs more data to train.

Detecting Collaboration using LSTM



- Encoder-decoder
  - The encoder converts the observation sequence to a single vector known as the final hidden state or context.
  - The decoder takes the final context vector as input and infers a collaborative attack.
  - Issue: the context becomes the ‘bottleneck’ since everything about the input sequence is encoded into a single vector.

Detecting Collaboration using LSTM with Attention Mechanism



- **Encoder-decoder with Attention**
  - Instead of taking the single final hidden state as the context for decoder input, the attention model takes a **weighted sum of all encoder's hidden states** as the context vector.

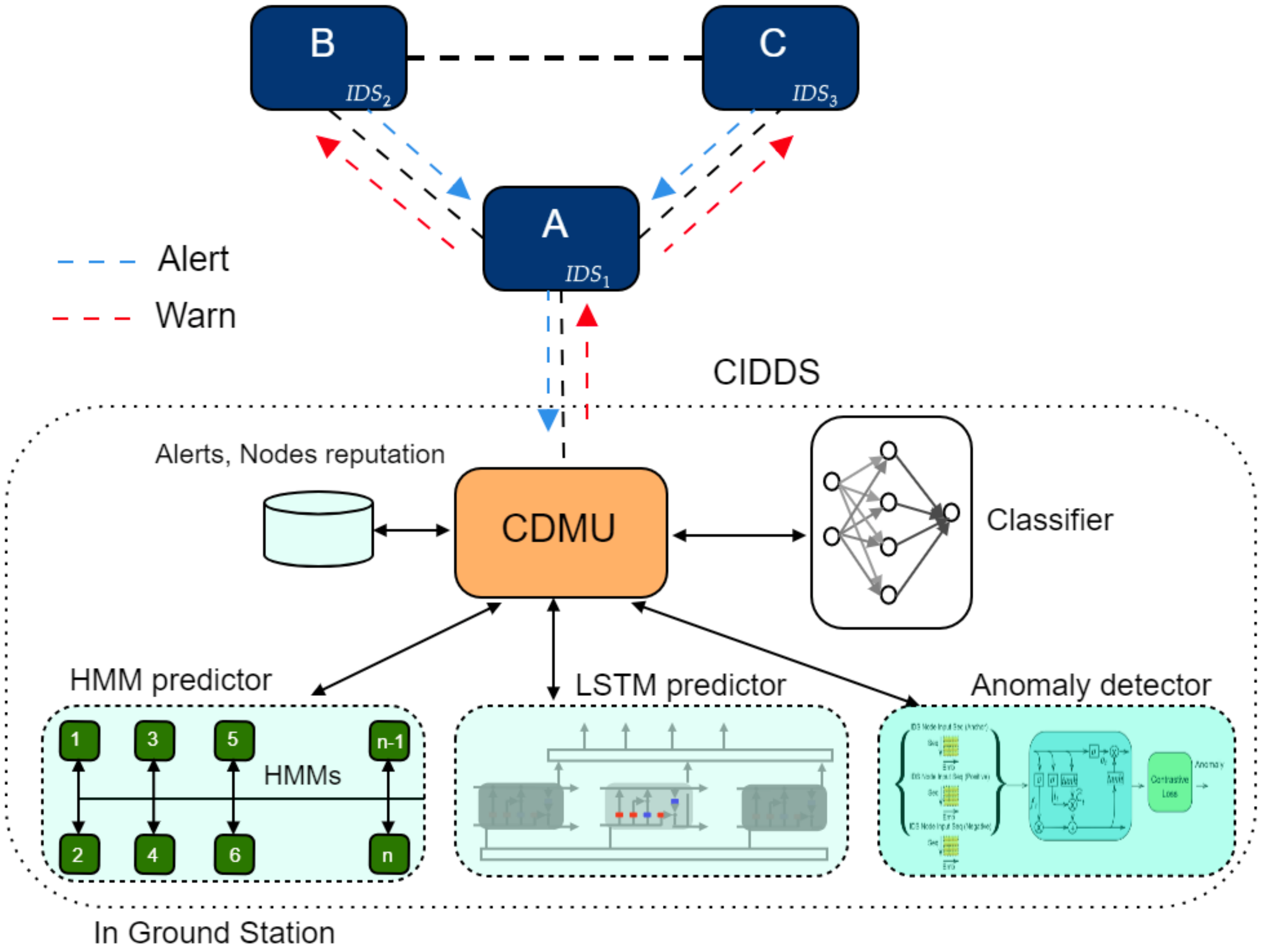
Anomaly Detection using Contrastive Learning

- The output of attention-based LSTM can be improved through a contrastive learning-based mechanism.
- This approach focuses on capturing meaningful relationships within a dataset (of proposed **alerts**) by **contrasting similar and dissimilar data points**.

Anomaly Detection using Contrastive Learning

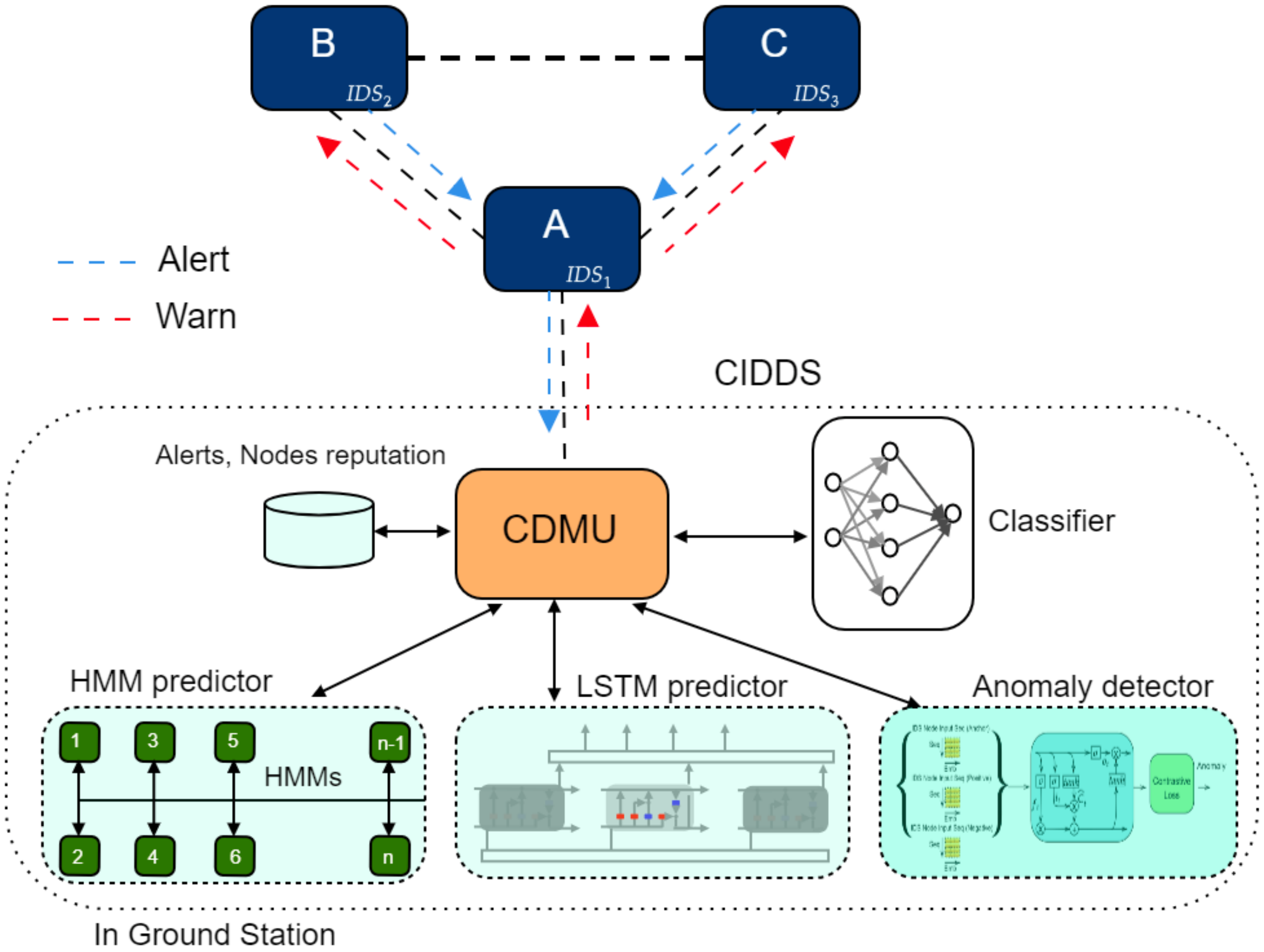
- The algorithm takes three types of samples, namely, **anchor (A), positive (P), and negative (N)** sets.
    - Anchor is a specific data sample (with anomaly) chosen as a point of comparison. The positive (P) data point is also an anomaly. The negative (N) is a legitimate data point.
- The algorithm **minimizes** the distance between **A and P** while **maximizing** the distance between **A and N**.
- The inference phase determines whether an input is positive or negative based on its distance to the anchor.
- In the context of attack detection, A and P could be anomaly data with the same or different attack types, while N is legitimate data.

Collaborative defense System



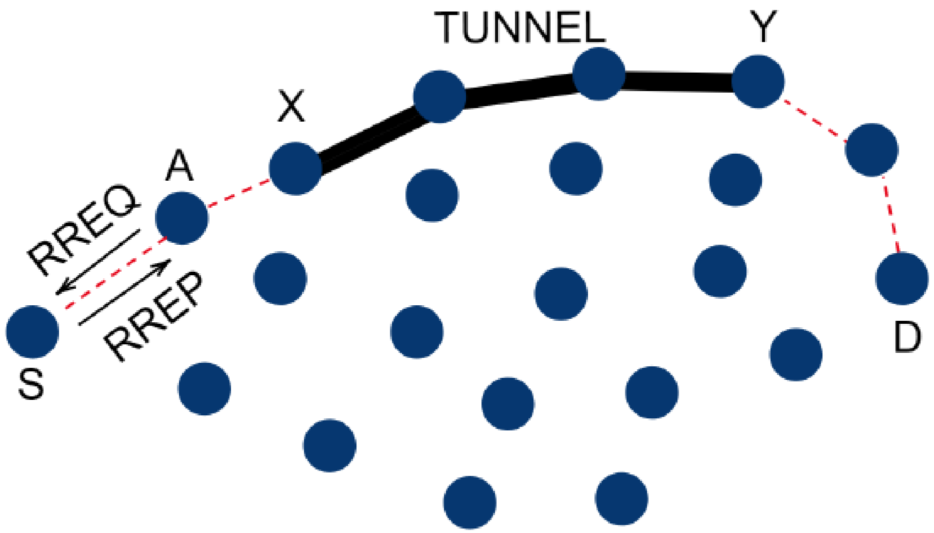
- Each node is equipped with Intrusion Detection System (**IDS**).
- IDS continuously monitors the nodes' security and generates alerts to **CIDDS**.
- CIDDS is the Collaborative Intrusion Detection and Defense System located at the **ground station**.
- CIDDS has a controller and decision-making unit (**CDMU**) to process and save the alerts locally
- CDMU also feeds alerts to the **HMM predictor, LSTM** predictor, and **anomaly** detector.

Collaborative defense System



- The outcome from the HMM, LSTM, and Anomaly, along with additional features related to the nodes' behavior (called here reputation) are properly conveyed by CMDU to the **deep learning classifier**.
- **Perceived ongoing collaborations for attacks are sent by the CDMU to every node in the network.**

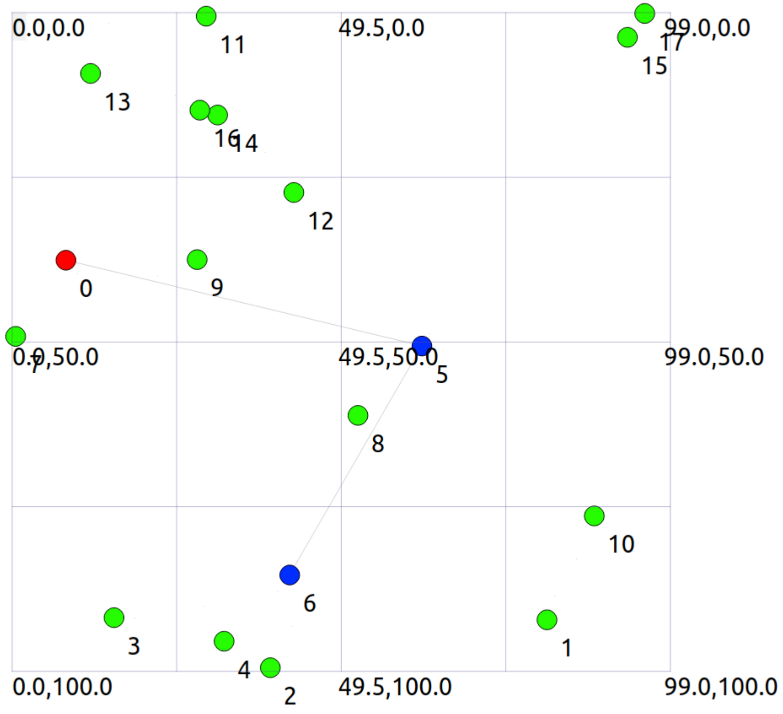
Wormhole and Blackhole



- Source: S and Destination: D
- Node A perpetrates a blackhole attack and nodes X and Y collude to carry out a wormhole attack.
- In this scenario, **if nodes A and X collaborate**, then the data packets from node S will be forwarded through the tunnel.

Implementation of collaborative attacks

- ns3 network simulator is used for attack implementation.
- Rich types of data can be collected from the simulator for attack detection and defense
  - Network traffic
  - (GPS) coordinates / trajectory
  - Routing table / topology information
  - Signal strength
- Right figure shows the implementation of wormhole and blackhole collaborative attack.
  - Red node is the blackhole attacker
  - Blue nodes are wormhole attackers
  - Gray lines are tunnel/wormhole



Problem Statement

Packet drop attacks put severe threats to Ad Hoc network performance and safety

- Directly impact the parameters such as packet delivery ratio
- Will impact security mechanisms such as distributed node behavior monitoring
- Different approaches have been proposed
  - Vulnerable to collaborative attacks
  - Have strong assumptions of the nodes

Problem Statement

Many research efforts focus on individual attackers

- The effectiveness of detection methods will be weakened under collaborative attacks
  - E.g., in “watchdog”, multiple malicious nodes can provide fake evidences to support each other’s innocence
  - In wormhole and Sybil attacks, malicious nodes may share keys to hide their real identities

Problem Statement

We focus on collaborative packet drop attacks. Why?

- Secure and robust data delivery is a top priority for many applications
- The proposed approach can be achieved as a reactive method: reduce overhead during normal operations
- Can be applied in parallel to secure routing

Related Work

Detecting packet drop attacks

- Audit based approaches
  - Whether or not the next hop forward the packets
  - Use both first hand and second hand evidences
  - Problems:
    - Energy consumption of eavesdropping
    - Can be cheated by directional antenna
    - Authenticity of the evidence
- Incentive based approaches
  - Nuggets and credits
- Multi-hop acknowledgement

Related Work

Collaborative attacks and detection

- Classification of the collaborative attacks
- Collusion attack model on secure routing protocols
- Collaborative attacks on key management in MANET
- Detection mechanisms:
    - Collaborative IDS systems
    - Ideas from immune systems
    - Byzantine behavior based detection

REAct system and Vulnerability

REAct system:

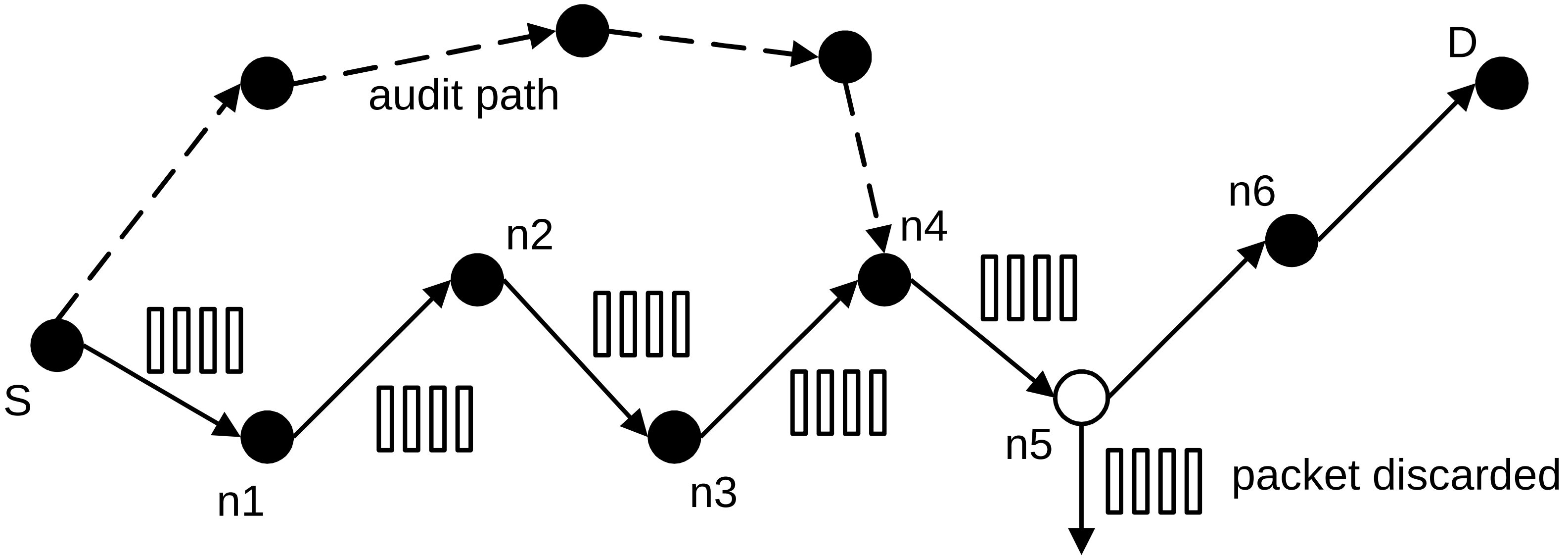
- Proposed by researchers in Arizona, ACM WiSec 2009
- Random audit based detector of packet drop
- A reactive approach: will be activated only when something bad happens
- Assumptions:
    - At least two node disjoint paths b/w any pair of nodes
    - Know the identity of the intermediate nodes
    - Pair-wise keys b/w the source and the intermediate nodes

REAct system and Vulnerability

Working procedure of REAct

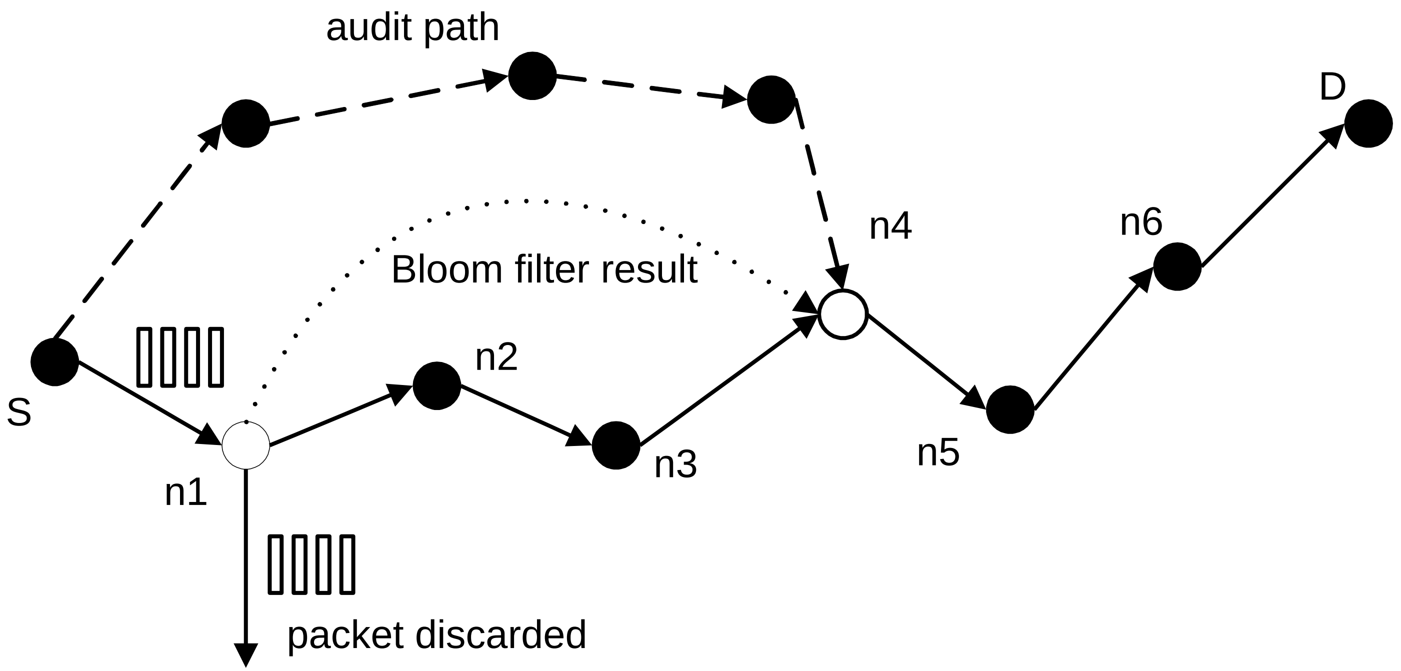
- Destination detects the drop in packet arriving rate and notifies the source
- Source randomly selects an intermediate node and asks it to generate a behavioral proof of the received packets
- Intermediate node constructs a bloom filter using these packets
- Source compares the bloom filter to its own value
  - If match: the attacker is after the intermediate node
  - Otherwise, it is before the intermediate node
- Repeat the procedure until the bad link is located

REAct system and vulnerability



Example of REAct: the source selects n4 to be the first audited node. n4 generates the correct bloom filter, so the attacker is between n4 and D.

Collaborative attacks on REAct



n1 and n4 are collusive attackers. n1 discards the packets but delivers the bloom filter to n4. Now the source will think that the attacker is between n4 and D.

Why REAct is vulnerable to this attack: the source can verify the bloom filter, but not the generator of the filter.

Proposed approach

Assumptions:

- Source shares a different secret key and a different random number with every intermediate node
- All nodes in the network agree on a hash function *h()*
- There are multiple attackers in the network
    - They share their secret keys and random numbers
    - Attackers have their own communication channel
    - An attacker can impersonate other attackers

Proposed approach

Hash based approach:

- Every node will add a fingerprint into the packet

  S1 sends out the packet to n1:

  S → *n1*: (*S*, *D*, data packet, random number *t0*)

  Node *n1* will combine the received packet and its random number *r1* to calculate the new fingerprint:

  t1 = h(*r1* || *S* || *D* || data packet || *t0* || *r1*)

  n1 → *n2*: (*S*, *D*, data packet, *t1*)

  The audited node will generate the bloom filter based on the data packets and the fingerprints

  The source will generate its own bloom filter and compare it to the value of the audited node

Proposed approach

Why our approach is safe

- The node behavioral proofs in our proposed approach contain information from both the data packets and the intermediate nodes.
- Theorem 1. If node *ni* correctly generates the value *ti*, then all innocent nodes in the path before *ni* (including *ni*) must have correctly received the data packet selected by *S*.

Proposed approach

Why this approach is safe

- The ordered hash calculations guarantee that any update, insertion, and deletion operations to the sequence of forwarding nodes will be detected.
- Therefore, we have:
  - if the behavioral proof passes the test of S, the suspicious set will be reduced to {*ni, ni+1, ---, D*}
  - if the behavioral proof fails the test of S, the suspicious set will be reduced to {S, n1, ---, ni}

Discussion

- Indistinguishable audit packets
  - The malicious node should not tell the difference between the data packets and audited packets
  - The source will attach a random number to every data packet
- Reducing computation overhead
  - A hash function needs 20 machine cycles to process one byte
  - We can choose a part of the bytes in the packet to generate the fingerprint. In this way, we can balance the overhead and the detection capability.

Discussion

- Security of the proposed approach
    - The hash function is easy to compute: very hard to conduct DoS attacks on our approach
    - It is hard for attackers to generate fake fingerprint: they have to have a non-negligible advantage in breaking the hash function
- The attackers will adjust their behavior to avoid detection
    - The source may choose multiple nodes to be audited at the same time
    - The source should adopt a random pattern to determine the audited nodes

Dealing with Collaborative Attacks

- Earlier approach is vulnerable to collaborative attacks
- Propose a new mechanism for nodes to generate behavioral proofs
  - Hash based packet commitment
  - Contain both contents of the packets and information of the forwarding paths
  - Introduce limited computation and communication overhead
- Extensions:
  - Investigate other collaborative attacks
  - Integrate our detection method with secure routing protocols