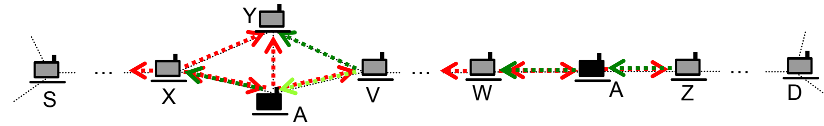
Provable security for ad hoc network routing protocols

motivation for a rigorous analysis framework;
simulation-based approach:
- real world model;
- ideal world model;
- definition of security;
- proof technique;
proof of endairA;

Provable security for ad hoc network routing protocols

- several "secure" routing protocols have been proposed for wireless ad hoc networks
  - SRP, Ariadne, S-AODV, ARAN, SEAD, …
- their security have been analyzed mainly by informal means
- informal reasoning about security protocols is prone to errors
  - lessons learnt in the field of key exchange protocols
  - some attacks have been found against SRP, Ariadne, and S-AODV
- we need more assurances
  - mathematical models
  - precise definitions
  - sound proof techniques

An attack on Ariadne



$X \rightarrow *$: [RREQ, S, D, id, h_X, (…, X), (…, mac_{XD})]
$A \rightarrow *$: [RREQ, S, D, id, *, (…, X, A), (…, mac_{XD}, h_X)]
…   …
$W \rightarrow *$: [RREQ, S, D, id, *, (…, X, A, V, …, W), (…, mac_{XD}, h_X, …, mac_{WD})]
A : $h_A = H(A \mid h_X)$
$A \rightarrow *$: [RREQ, S, D, id, h_A, (…, X, A), (…, mac_{XD}, mac_{AD})]
…   …
$Z \rightarrow A$: [RREP, D, S, (…, X, A, Z, …), mac_{DS}]
$A \rightarrow W$: [RREP, D, S, (…, X, Y, V, … W, A, …), mac_{DS}]
…   …
$V \rightarrow Y$: [RREP, D, S, (…, X, Y, V, … W, A, …), mac_{DS}]
$A \rightarrow X$: [RREP, D, S, (…, X, A, Z, …), mac_{DS}]
…   …
$? \rightarrow S$: [RREP, D, S, (…, X, A, Z, …), mac_{DS}]   **(a non-existent route!)**

Mathematical framework

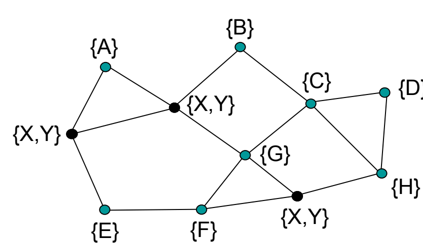
- based on the simulation paradigm
  - real-world model
    - describes the real operation of the protocol
  - ideal-world model
    - captures what the protocol wants to achieve in terms of security
  - definition of security in terms of indistinguishability of the two models from the point of view of honest participants

Mathematical framework (cont'd)

- communication model
  - multi-hop communication and the broadcast nature of radio channels are explicitly modeled
- adversary model
  - power of the adversary is limited
  - it has communication capabilities similar to regular nodes
  - it cannot fully control when some nodes send and receive messages
- model of computation
  - computation is not scheduled by the adversary
  - computation is performed in rounds (synchronous model), but …
  - knowledge of the current round number is never exploited
- ideal-world model and ideal-world adversary
  - they are essentially the same as the real-world model and adversary
  - the ideal world is ideal in the following sense:
    - route reply messages that contain incorrect routes are marked and filtered out
    - incorrect routes are never returned in the ideal world

Configuration

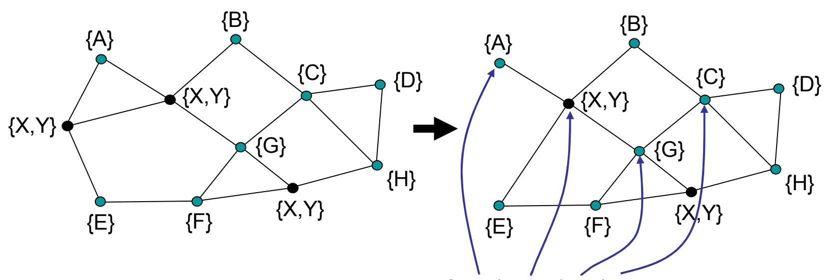
- an ad hoc network is represented by a graph $G(V, E)$
  - V: vertices are network nodes (honest and adversarial)
  - E: edges represent communication links (radio or wormhole)
- $V^* \subset V$ is a set of distinguished nodes (under the adversary's control)
- $\mathcal{L}$ is a labeling function (assigns IDs to nodes) with the following restrictions:
  - each honest node has a unique, uncompromised ID
  - each adversarial node is labeled with *all* the compromised IDs
  - we assume that ID's are authenticated during neighbor discovery (Sybil attack is excluded)



- a *configuration* is a triplet: $(G, V^*, \mathcal{L})$

Plausible routes

- reduced configuration: $(\underline{G}(\underline{V}, \underline{E}), \underline{V}^*, \underline{\mathcal{L}})$
  - neighboring adversarial nodes are joined
- a route is *plausible* in a given configuration, if it doesn't contain repeating IDs and it can be partitioned in a way that each partition P can be associated with a node v in $\underline{G}$ such that
  - $P \subseteq \underline{\mathcal{L}}(v)$, and
  - neighboring partitions are associated with neighboring nodes in $\underline{G}$



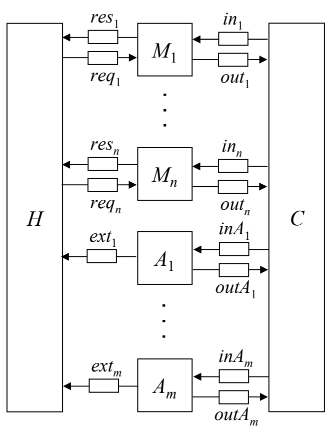
A X Y G C → A | X Y | G | C

A X G D H → non-plausible

The rational behind plausible routes

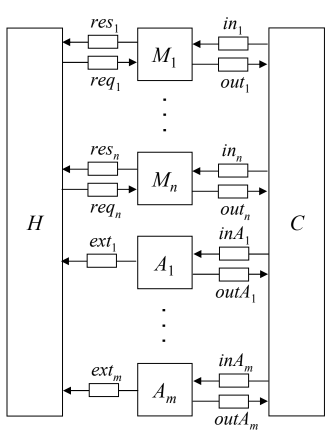
- adversarial nodes can emulate the execution of the routing protocol (locally) using any subset of the compromised IDs in any order
- they can also pass information to each other in a proprietary way
- these are *tolerable imperfections*, which are embedded in the notion of plausible routes

Real-world model (1)



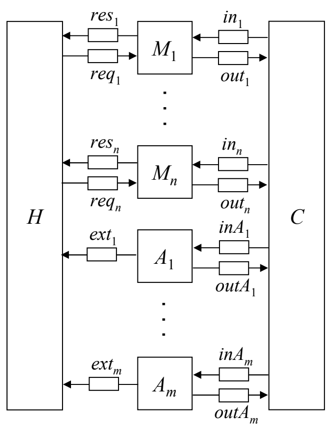
- $H, M_1, \ldots, M_n, A_1, \ldots, A_m, C$ are interacting, probabilistic Turing machines
  - $M_1, \ldots, M_n$ represent honest nodes in G
  - $A_1, \ldots, A_m$ represent adversarial nodes in G
  - C models the communication links (edges of G)
- each machine is initialized with some input data (e.g., crypto keys) and some random input
- each machine operates in a reactive manner (must be activated)
  - reads input tape
  - performs state transition and writes output tape
  - goes back to sleep
- machines are activated by a hypothetic scheduler in rounds in a fix order in each round: $H, \ldots, C$
- the computation ends when H reaches a final state

Real-world model (2)



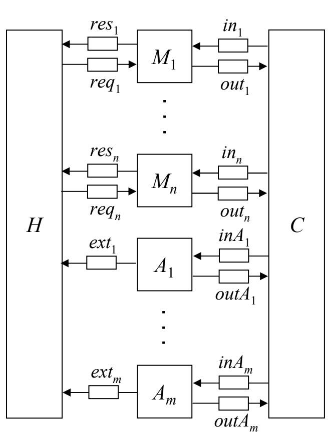
- C models the communication links
  - when activated, it moves the content of the output tape of each protocol machine (M_i and A_j) onto the input tape of all neighboring machines in G (in a random order)
- H models higher layer protocols (and ultimately the end-users) of non-corrupted nodes
  - it can initiate a route discovery process at any machine M_i by placing a request on req_i
  - a response may be returned to the request via res_i
  - the response contains a set of routes (maybe empty set)
  - it can receive out-of-band requests from the adversarial machines via ext_j

Real-world model (3)



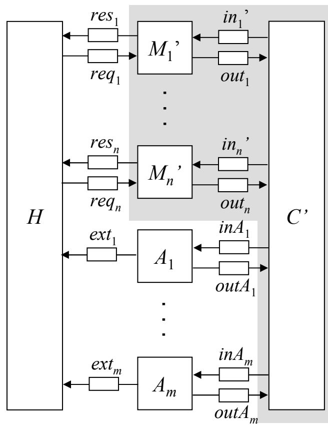
- M_i models the operation of the routing algorithm in the i-th non-corrupted node
  - it receives requests from H via req_i and may return a response via res_i
  - it sends and receives routing messages to and from its neighbors via out_i and in_i
  - initialized with its own ID and those of its neighbors, some cryptographic material, and random input
- A_j models the j-th adversarial node
  - it uses $outA_j$ and inA_j to communicate with its neighbors
  - it can use ext_j to "force" H to start a route discovery between any two *honest nodes*
  - it is *non-adaptive*: it places its requests on ext_j at the beginning of the computation, and doesn't use ext_j anymore
  - its behavior is not restricted apart from being polynomial-time in the security parameter

Real-world model (4)



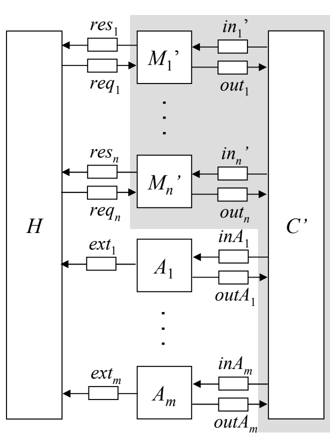
- output of the real-world model
  - sets of routes returned to H
  - denoted by $real\_out_{conf,\mathcal{A}}(r)$, where $r = (r_I, r_M, r_A, r_C)$
    - r_I – random input of cryptographic initialization (key generation)
    - r_M – random input of $M_1, \ldots, M_n$
    - r_A – random input of $A_1, \ldots, A_m$
    - r_C – random input of C
  - $real\_out_{conf,\mathcal{A}}$ denotes the random variable describing the output when r is chosen uniformly at random

Ideal-world model (1)



- difference between C and C':
  - C' marks every route reply message that contains a non-plausible route as corrupted before placing it on the input tape in_i' of a non-corrupted protocol machine M_i
  - otherwise C' works in the same way as C
- difference between M_i and M_i':
  - when M_i' receives a route reply message that belongs to a route discovery process initiated by itself, it processes the message as follows:
    - it performs all the verifications required by the routing protocol
    - if the message passes all verifications, then it also checks the corruption flag attached to the message
    - if the message is corrupted (contains a non-plausible route), then M_i' drops the message
  - otherwise M_i' behaves as M_i

Ideal-world model (2)



- output of the ideal-world model
  - sets of routes returned to H
  - denoted by $ideal\_out_{conf,\mathcal{A}}(r')$, where $r' = (r'_I, r'_M, r'_A, r'_C)$
  - $ideal\_out_{conf,\mathcal{A}}$ denotes the random variable describing the output when r' is chosen uniformly at random

Definition of (statistical) security

A routing protocol is said to be (statistically) secure if, for any configuration $conf$ and any real-world adversary $\mathcal{A}$, there exists and ideal-world adversary $\mathcal{A}$', such that

$$real\_out_{conf,\mathcal{A}} =^{s} ideal\_out_{conf,\mathcal{A}'}$$

where $=^{s}$ means statistically indistinguishable.

<u>notes</u>:

- two random variables are statistically indistinguishable if the L_1 distance of their distributions are negligibly small
- if this definition is satisfied by a protocol, then a non-plausible route can be returned in the real system only with negligible probability (for every configuration and arbitrary adversary)

Proof technique

- let $\mathcal{A}' = \mathcal{A}$
- if, for a given r, no message is dropped due to its corruption flag in the ideal-world model, then the ideal-world model perfectly simulates the real-world model:

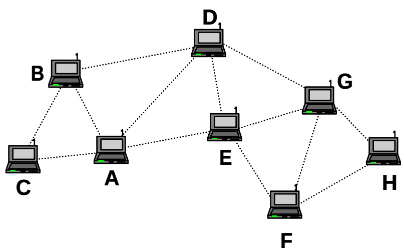
$$real\_out_{conf,\mathcal{A}}(r) = ideal\_out_{conf,\mathcal{A}}(r)$$

- if, for some r, there exist messages that are dropped due to their corruption flag in the ideal-world model, then there may be a *simulation failure*:

$$real\_out_{conf,\mathcal{A}}(r) \neq ideal\_out_{conf,\mathcal{A}}(r)$$

- in proofs, we want to show that simulation failures occur with negligible probability
- if this is not the case, then
  - in theory, we haven't proven anything (there may be another $\mathcal{A}' \neq \mathcal{A}$, for which we have statistical indistinguishability)
  - in practice, there's a problem with the protocol

Reminder on endairA



target verifies:
- there's no repeating ID in the node list
- last node in the node list is a neighbor

each intermediate node verifies:
- its own ID is in the node list
- there's no repeating ID in the node list
- next and previous nodes in the node list are neighbors
- all signatures are valid

source verifies:
- there's no repeating ID in the node list
- first node in the node list is a neighbor
- all signatures are valid

A → * : [RREQ, A, H, id, ()]
E → * : [RREQ, A, H, id, (E)]
F → * : [RREQ, A, H, id, (E, F)]

H → F : [RREP, A, H, id, (E, F), (sig_H)]
F → E : [RREP, A, H, id, (E, F), (sig_H, sig_F)]
E → A : [RREP, A, H, id, (E, F), (sig_H, sig_F, sig_E)]

Analysis of endairA (1)

Theorem:

endairA is statistically secure if the signature scheme is secure against chosen message attacks.

sketch of the proof:

- it is enough to prove that, for any configuration *conf* and attacker $\mathcal{A}$, a route reply message in the ideal-world system is dropped due to its corruption flag set to true with negligible probability
- let us suppose that the following message is dropped due to its corruption flag:

$$[\text{ RREP}, S, D, (N_1, N_2, \ldots, N_p), (sig_D, sig_{N_p}, \ldots, sig_{N_1})]$$

- we know that
  - there are no repeating IDs in $(S, N_1, N_2, \ldots, N_p, D)$
  - N_1 is a neighbor of S
  - all signatures are valid
  - S and D are honest
  - $(S, N_1, N_2, \ldots, N_p, D)$ is a non-plausible route in $\underline{G}$
- we prove that $\mathcal{A}$ must have forged a signature to achieve this

Analysis of endairA (2)

sketch of the proof (cont'd):

- in the reduced configuration adversarial nodes are non-adjacent
- thus each sequence of non-repeating IDs has a unique partitioning
  - IDs of honest nodes form distinct partitions
  - consecutive adversarial IDs form a partition
- if the route is non-plausible, then (at least) one of the following must hold:
  - $P_j=\{N_i\}$ and $P_{j+1}=\{N_{i+1}\}$ are non-adversarial partitions and the nodes v and v' that belong to N_i and N_{i+1} are not adjacent in $\underline{G}$
  - $P_j=\{N_i\}$, $P_{j+1}=\{N_{i+1}, \ldots, N_{i+k}\}$, $P_{j+2}=\{N_{i+k+1}\}$ are two non-adversarial (P_j, P_{j+2}) and an adversarial partition (P_{j+1}) and the nodes that belong to N_i and N_{i+k+1} have no common neighbor that belongs to V^*
- in the first case, the node that uses N_i would detect that the next ID in the list doesn't belong to a neighbor and wouldn't sign the message

Analysis of endairA (3)

- **sketch of the proof (cont'd):**
  - **in the second case:**
    - assume the adversary didn't forge any signatures
    - the node using N_i must have received

      [RREP, S, D, $(N_1, N_2, \ldots, N_p)$, $(sig_D, sig_{Np}, \ldots, sig_{N_{i+1}})$]

      from an adversarial node, say A (why?)
    - A must have received

      [RREP, S, D, $(N_1, N_2, \ldots, N_p)$, $(sig_D, sig_{Np}, \ldots, sig_{N_{i+k+1}})$]

      from N_{i+k+1} (why?)
    - A must be a common neighbor of N_i and N_{i+k+1}, which is a contradiction

  → the adversary must have forged some signatures

Summary

- attacks against secured ad hoc network routing protocols exist
- flaws are subtle and difficult to discover by informal analysis
- the simulation-based analysis approach used in cryptography can be adopted for reasoning about the security of ad hoc network routing protocols
  - we showed this for on-demand source routing protocols, but the same ideas work for other types of protocols too
- unfortunately, hand-written proofs are tedious and prone to errors
- open question: How to automate the case analysis in proofs?

Security and Privacy in Upcoming Wireless Networks

Wormhole detection

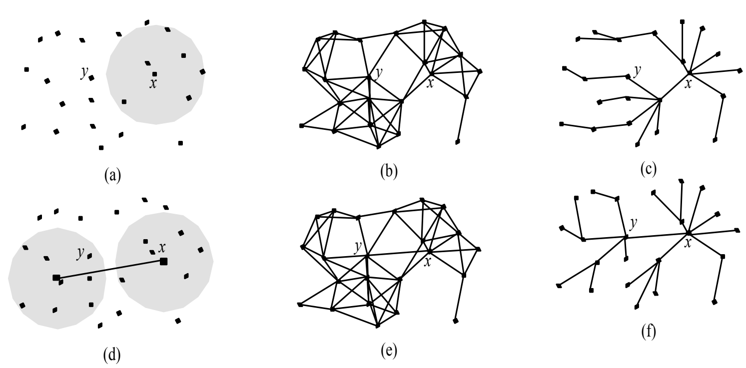
the wormhole attack;
centralized and decentralized wormhole detection mechanisms;

What is a wormhole?

- a wormhole is an out-of-band connection, controlled by the adversary, between two physical locations in the network
  - the adversary installs radio transceivers at both ends of the wormhole
  - it transfers packets (possibly selectively) received from the network at one end of the wormhole to the other end via the out-of-band connection, and re-injects the packets there into the network
- notes:
  - adversary’s transceivers are not regular nodes (no node is compromised by the adversary)
  - adversary doesn’t need to understand what it tunnels (e.g., encrypted packets can also be tunneled through the wormhole)
  - it is easy to mount a wormhole, but it may devastating effects on routing

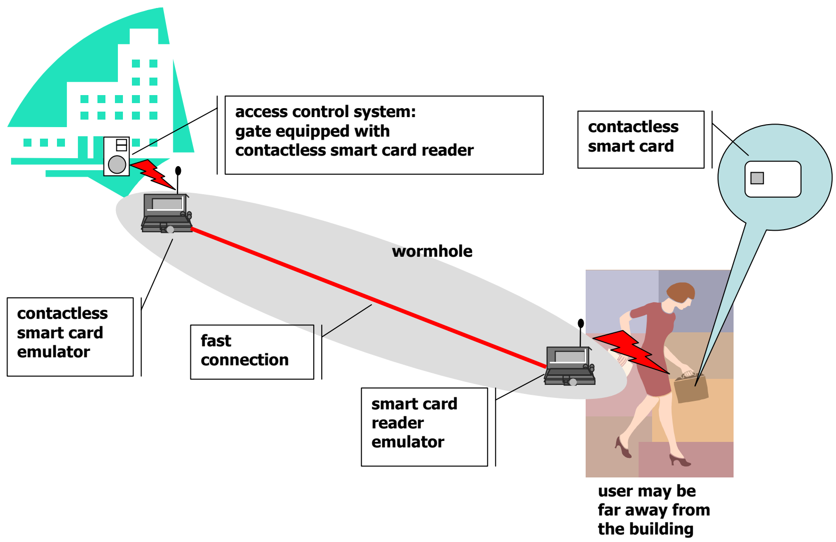
Effects of a wormhole

- at the data link layer: distorted network topology



- at the network layer:
  - routing protocols may choose routes that contain wormhole links
    - typically those routes appear to be shorter
    - flooding based routing protocols (e.g., DSR, Ariadne) may not be able to discover other routes but only through the wormhole
  - adversary can then monitor traffic or drop packets (DoS)

Wormholes are not unique to ad hoc networks



Classification of wormhole detection methods

- centralized mechanisms
  - data collected from the local neighborhood of every node are sent to a central entity
  - based on the received data, a model of the entire network is constructed
  - the central entity tries to detect inconsistencies (potential indicators of wormholes) in this model
  - can be used in sensor networks, where the base station can play the role of the central entity
- decentralized mechanisms
  - each node constructs a model of its own neighborhood using locally collected data
  - each node tries to detect inconsistencies on its own
  - advantage: no need for a central entity (fits well some applications)
  - disadvantage: nodes need to be more complex

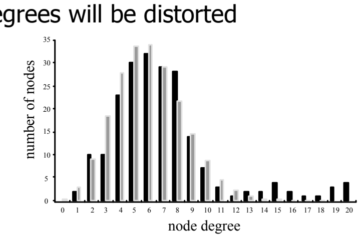
Statistical wormhole detection in sensor networks

- each node reports its list of believed neighbors to the base station
- the base station reconstructs the connectivity graph (model)
- *a wormhole always increases the number of edges* in the connectivity graph
- this increase may change the properties of the connectivity graph in a detectable way (anomaly)
- detection can be based on statistical hypothesis testing methods (e.g. the χ^2-test)

Examples

- a wormhole that creates many new edges may increase the *number of neighbors* of the affected nodes

→ distribution of node degrees will be distorted



- a wormhole is usually a shortcut that decreases the length of the shortest paths in the network

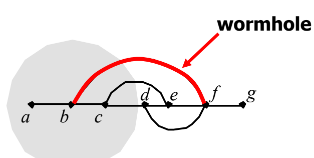
→ distribution of the length of the shortest paths will be distorted

Multi-dimensional scaling

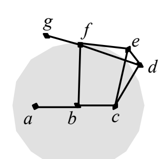
- the nodes not only report their lists of neighbors, but they also estimate (inaccurately) their distances to their neighbors
- connectivity information and estimated distances are input to a multi-dimensional scaling (MDS) algorithm
- the MDS algorithm tries to determine the possible position of each node in such a way that the constraints induced by the connectivity and the distance estimation data are respected
  - the algorithm has a certain level of freedom in "stretching" the nodes within the error bounds of the distance estimation
- let us suppose that an adversary installed a wormhole in the network
  - if the estimated distances between the affected nodes are much larger than the nodes' communication range, then the wormhole is detected
  - hence, the adversary must also falsify the distance estimation → distances between far-away nodes become smaller
  - this will result in a distortion in the virtual layout constructed by the MDS algorithm

Examples

- in 1D:

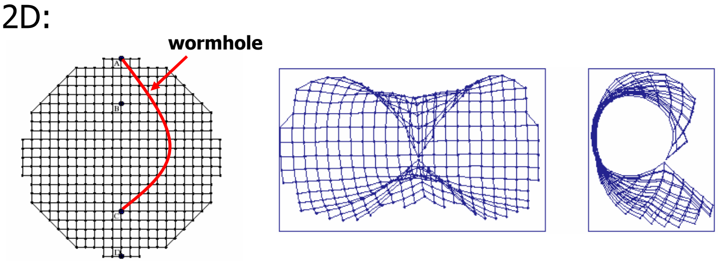


connectivity graph



reconstructed virtual layout

- in 2D:



Packet leashes

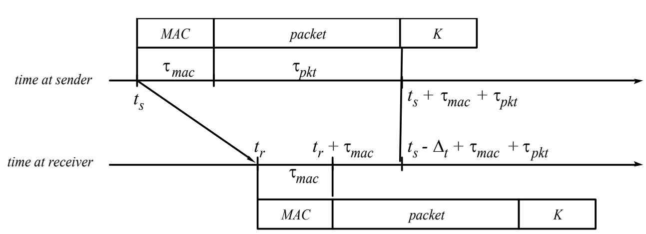
- packet leashes ensure that packets are not accepted "too far" from their source
- geographical leashes
  - each node is equipped with a GPS receiver
  - when sending a packet, the node puts its GPS position into the header
  - the receiving node verifies if the sender is really within communication range
- temporal leashes
  - nodes' clocks are very tightly synchronized
  - when sending a packet, the node puts a timestamp in the header
  - the receiving node estimates the distance of the sender based on the elapsed time and the speed of light

$$d_{est} < v_{light}(t_{rcv} - t_{snd} + \Delta_t)$$

  - note: $v_{light}\,\Delta_t$ must be much smaller than the communication range

TESLA with Instant Key-disclosure (TIK)

<u>idea</u>: authentication delay of TESLA can be removed in an environment where the nodes' clocks are tightly synchronized



- by the time the sender reveals the key, the receiver has already received the MAC
- security condition: $t_r < t_s - \Delta_t + t_{pkt}$
- note: Δ_t must be very small or otherwise packets must be very long

Mutual Authentication with Distance-bounding (MAD)

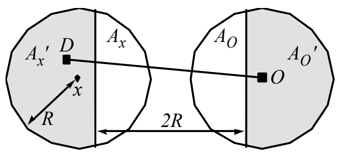
u		v
	— initialization phase —	
generate random numbers $r \in \{0,1\}^{\ell}$, $r' \in \{0,1\}^{\ell'}$		generate random numbers $s \in \{0,1\}^{\ell}$, $s' \in \{0,1\}^{\ell'}$
compute commitment $c_u = H(r\|r')$		compute commitment $c_v = H(s\|s')$
	$\xrightarrow{c_u}$	
	$\xleftarrow{c_v}$	
	— distance-bounding phase —	
the bits of r are $r_1, r_2, \ldots, r_\ell$		the bits of s are $s_1, s_2, \ldots, s_\ell$
$\alpha_1 = r_1$	$\xrightarrow{\alpha_1}$	
	$\xleftarrow{\beta_1}$	$\beta_1 = s_1 \oplus \alpha_1$
	$\ldots$	
$\alpha_i = r_i \oplus \beta_{i-1}$	$\xrightarrow{\alpha_i}$	measure delay between β_{i-1} and α_i
measure delay between α_i and β_i	$\xleftarrow{\beta_i}$	$\beta_i = s_i \oplus \alpha_i$
	$\ldots$	
$\alpha_\ell = r_\ell \oplus \beta_{\ell-1}$	$\xrightarrow{\alpha_\ell}$	measure delay between $\beta_{\ell-1}$ and α_ℓ
measure delay between α_ℓ and β_ℓ	$\xleftarrow{\beta_\ell}$	$\beta_\ell = s_\ell \oplus \alpha_\ell$
	— authentication phase —	
$s_i = \alpha_i \oplus \beta_i$ $(i = 1, \ldots, \ell)$		$r_1 = \alpha_1$ and $r_i = \alpha_i \oplus \beta_{i-1}$ $(i = 2, \ldots, \ell)$
$\mu_u = mac_{k_{uv}}(u\|v\|r_1\|s_1\|\ldots\|r_\ell\|s_\ell)$		$\mu_v = mac_{k_{uv}}(v\|u\|s_1\|r_1\|\ldots\|s_\ell\|r_\ell)$
	$\xrightarrow{r'\|\mu_u}$	
	$\xleftarrow{s'\|\mu_v}$	
verify c_v and μ_v		verify c_u and μ_u

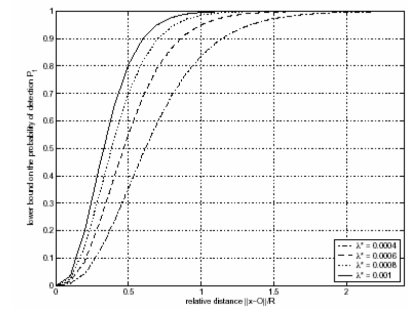
- MAD allows precise distance estimation without synchronized clocks

Using position information of anchors

- anchors are special nodes that know their own positions (GPS)
- there are only a few anchors randomly distributed among regular nodes
- two nodes consider each other neighbors only if
  - they hear each other and
  - they hear more than T common anchors
- anchors put their location data in their messages
- transmission range of anchors (R) is larger than that of regular nodes (r)
- wormholes are detected based on the following two principles:
  1. a node should not hear two anchors that are 2R apart from each other
  2. a node should not receive the same message twice from the same anchor

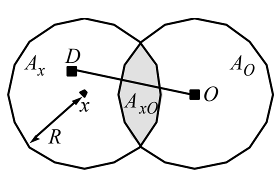
Principle 1

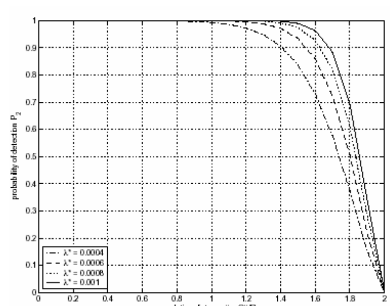




- x hears anchors in A_x and in A_O
- P_1 is the probability that it hears two anchors that are further away from each other than 2R
- the probability that there is at least one anchor in an area of size S is $(1-e^{-\lambda^* S})$, where λ^* is the density of anchors
- $P_1 \geq (1-e^{-\lambda^* S'_x})(1-e^{-\lambda^* S'_O})$, where S'_x is the size of A'_x and S'_O is the size of A'_O
- this lower bound is maximum when $S'_x = S'_O$

Principle 2

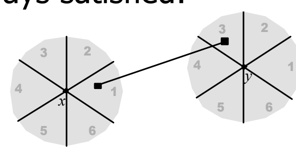




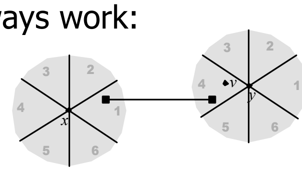
- when x and O are closer than 2*R*, the discs A_x and A_O overlap
- if there is an anchor in the intersection A_{xO}, then the messages of that anchor is heard twice by *x*
  - first directly and then from transceiver D who receives it from O through the wormhole
- the probability P_2 of detection is equal to the probability that there is at least one anchor in A_{xO}
- $P_2 = 1-e^{-\lambda^* S_{xO}}$

Wormhole detection with directional antennas

- when two nodes are within each other's communication range, they must hear each other's transmission from opposite directions
- if nodes x and y communicate through a wormhole, then this condition is not always satisfied:



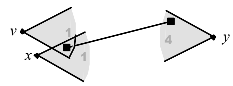
- but this doesn't always work:



Using verifiers

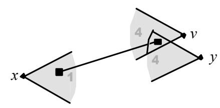
- verifiers are common neighbors satisfying certain conditions
- y accepts x as a neighbor if
  - they hear each other from opposite zones
  - there's at least one ***valid verifier*** v such that x and v hear each other from opposite zones

Conditions for being a valid verifier



- if node y hears v in the same zone in which it hears x, then y may hear both x and v through the wormhole

→ for a valid verifier v, y must hear v and x from different zones (i.e., $Z_{yv} \neq Z_{yx}$ must hold)



- if v hears x in the same zone in which y hears x (i.e., $Z_{vx} = Z_{yx}$), then they may both hear x through the wormhole's transceiver
- if, in addition, x happens to hear the other transceiver of the wormhole in zone $\underline{Z}_{yx}$, then x can establish neighbor relationships with both y and v

→ for a valid verifier v, v must hear x from a zone different from the one in which y hears x (i.e., $Z_{vx} \neq Z_{yx}$ must hold too).

How does this detect wormholes?

- let us assume that y hears x through the wormhole
  - → one end of the wormhole is near to x, the other end is in zone Z_{yx}
- let us further assume that v is a valid verifier

→ first condition ($Z_{yv} \neq Z_{yx}$) is satisfied
  - → y hears v directly (since y hears v from a zone different from Z_{yx})
  - → x hears both y and v through the wormhole

→ second condition ($Z_{vx} \neq Z_{yx}$) is satisfied
  - → x and v cannot hear each other from opposite zones
    - let's assume that $Z_{xv} = \underline{Z}_{vx}$
    - we know that x hears both y and v through the wormhole → $Z_{xy} = Z_{xv}$
    - in addition, we know that $Z_{xy} = \underline{Z}_{yx}$ (otherwise y would not consider x as a potential neighbor)
    - $\underline{Z}_{vx} = Z_{xv} = Z_{xy} = \underline{Z}_{yx}$ → $Z_{vx} = Z_{yx}$ (contradicts the second condition)

→ no valid verifier v exists such that x and v hear each other from opposite zones → y will not accept x as a neighbor

Summary

- a wormhole is an out-of-band connection, controlled by the adversary, between two physical locations in the network
- a wormhole distorts the network topology and may have a profound effect on routing
- wormhole detection is a complicated problem
  - centralized and decentralized approaches
    - statistical wormhole detection
    - wormhole detection by multi-dimensional scaling and visualization
    - packet leashes
    - distance bounding techniques
    - anchor assisted wormhole detection
    - using directional antennas
  - many approaches are based on strong assumptions
    - tight clock synchronization
    - GPS equipped nodes
    - directional antennas
    - …
- wormhole detection is still an active research area