



# CS649

## Sensor Networks

### Lectures 31-32: Security

Andreas Terzis

<http://hinrg.cs.jhu.edu/wsn06/>

# Overview

- Overview of (in)security in WSN routing protocols
  - Differences between WSNs and ad-hoc networks
  - Assumptions
  - General attacks
  - Attacks against existing sensor network protocols
  - Countermeasures
- Link Layer Security: TinySec
- SPINS

# WSNs vs. Ad-hoc wireless networks

- Traffic Patterns
  - Ad-hoc: any-to-any
  - WSNs:
    - Many-to-one
    - One-to-many
    - Localized communication
- WSNs have tighter resource constraints
- Trust relationships between nodes in a WSN are higher

# Assumptions

- Radio links are insecure
  - Attackers can eavesdrop on radio transmissions
  - Inject packets
  - Replay previously overheard packets
- Attacker is capable of deploying “a few” node with mote-compatible HW
  - Attacking nodes might be colluding
  - Attacking nodes might have high-capacity links between them (e.g. directional antennas)
- Sensor nodes are not *tamper-resistant*
  - Attacker can retrieve key material from compromised node
- Physical and MAC layers are susceptible to direct attacks

# Trust requirements

- Base station is *trustworthy*
  - If base station is not trustworthy then no information can be passed to the wide-area network
- Aggregation points (e.g. cluster-heads) are not trustworthy
  - They are just regular sensor nodes

# Threat Models

- Attacker Capabilities
  - Mote-class attacker
    - Attacker nodes have capabilities similar to motes
  - Laptop-class attacker
    - Attacker nodes have lower resource constraints
    - More power, more memory, high-power radio
- Trust
  - Outsider attacker
    - No special access to the sensor network
  - Insider attacker
    - Authorized participant of the sensor network

# Security Goals

- *Ideally* would like to guarantee
  - *Confidentiality*
  - *Integrity*
  - *Authenticity*
  - *Availability*
- Which of the above is the responsibility of the routing protocol?
  - WSNs differ from other networks due to *in-network processing*
- If the goals above are not attainable then *graceful degradation* is desired

# Attacks on WSN Routing

- Spoofed, altered, or replayed routing information
- Selective forwarding
- Sinkhole attacks
- Sybil attacks
- Wormholes
- HELLO flood attacks
- ACK spoofing

# Spoofed routing information

- By spoofing, altering, or replaying routing information adversary can:
  - Create routing loops
  - Attract or repel traffic
  - Extend or shorten routes
  - Generate false error messages
  - Partition the network
  - Etc...

# Selective Forwarding

- Malicious node refuses to forward certain messages
  - *Black hole*
  - *Selective forwarding*
  - Attack works when malicious node is on the path between source and destination
- Overhearing attacker can potentially create similar damage
  - Jamming
  - Collision

# Sinkhole

- Adversary's goal is to lure traffic from network area through a compromised node
  - Enables other attacks
  - Attack works by making compromised node *attractive*
- WSNs susceptible to this attack due to many-to-one traffic pattern

# Sybil Attack

- Single node presents multiple identities to the other nodes in the network
  - Such attacks can reduce the effectiveness of fault-tolerant schemes
  - Threat to geographic routing algorithms
    - Attacker presents multiple identities with multiple locations to neighbors

# Wormholes

- Adversary tunnels messages from one part of the network to another
  - Through low-latency, long-range private link between colluding malicious nodes
- Wormholes can create *routing race conditions*
  - Affect protocols where nodes take action based on first message they receive
- Wormholes can be used to convince two distant nodes that are neighbors
- Can be used in combination of selective forwarding or eavesdropping

# HELLO Flood Attack

- Many protocols use HELLO messages where node announces itself to its neighbors
  - Receiver assumes that sender is within range
  - False if sender has high-power radio
- Does not need the attacker to construct legitimate traffic
  - Can simply replay with high power legitimate overheard message

# ACK Spoofing

- WSN Routing algorithms rely on implicit or explicit ACKs
  - Adversary can *spoof* ACKs
    - Weak link is strong
    - Dead node is alive

# Attacks on Specific Protocols

- TinyOS Beaconsing
- Directed Diffusion
- Geographic Routing
- Minimum Cost Forwarding
- LEACH
- GEC
- SPAN, GAF

# TinyOS Beaconing

- TinyOS beaconing constructs breadth-first spanning tree routed at the base station
- Attacks
  - Malicious node claims to be the base station
  - Wormhole/Sinkhole attack
    - Works even with authenticated routing updates
  - HELLO flood attack
    - Every node in the network hears the announcement but cannot reach back
  - Routing loops

# Directed Diffusion

- Operation reminder
  - Base station floods interests setting up gradients
  - Nodes return data on reverse path
  - Paths are positively or negatively reinforced
- Attacks
  - Suppression: Attacker sends spoofed negative reinforcements
  - Cloning: Attacker replays interests from legit base stations (eavesdropping)
  - Path Influence: By spoofing positive and negative reinforcements
  - Selective forwarding and data tampering: By using the above technique to steer data through malicious node
  - Multi-path version is susceptible to Sybil attack

# Geographic Routing

- GPSR, GEAR use variants of greedy geographic routing
  - GEAR tries to load-balance traffic based on remaining node power
- Attacks
  - Attacker can announce arbitrary location to inject herself on the path of desired flow (for GEAR advertise max power)
  - Sybil attack

# Minimum Cost Forwarding

- Idea
  - Create a *cost field*
    - Base station advertises with cost 0
    - Node  $n$  re-advertises announcement from  $m$  with cost  $C_m + L_{m,n}$
  - Packets are sent upstream with *cost budget* initialized to the source min cost
  - Upstream node forwards packet only if remaining budget is equal to own min cost
- Attacks:
  - Sinkhole: Advertise zero cost
  - HELLO flood attack

# LEACH

- Reminder:
  - Adaptive clustering
  - Cluster-head can reach base station through second radio
  - Cluster-heads are rotated to balance power consumption
  - TDMA within the cluster
- Attacks
  - HELLO flood attack causes many nodes to pick laptop-attacker as their cluster-head
    - Countermeasures: refuse same cluster-head for multiple rounds, probabilistic selection of cluster-head
    - Can be subverted with Sybil attack

# SPAN, GAF

- GAF Reminder
  - Nodes are put in grid, nodes in adjacent grid squares can communicate
  - Three states: *sleeping*, *discovery*, and *active*
  - Attempt to reach state where 1 active node/square
- Attack
  - Adversary broadcast *high-rank* messages to force other nodes to sleep
- SPAN Reminder
  - Coordinators stay on to forward messages
  - Other nodes periodically probe to test whether they should become coordinators
- Attack
  - Attacker can announce fake list of coordinators

# Countermeasures

- Outsider attacks and link-layer security
  - Majority of attacks can be prevented with link-layer encryption and authentication using a globally shared key
  - Link-layer security does not protect against Wormhole and HELLO flood attacks
    - Attacker can simply replay messages from one side of the network to another or amplify overheard messages
- Link-layer security is completely ineffective in the presence of insider attacks or compromised nodes



# TinySec: A Link Layer Security Architecture for Sensor Networks

C. Karlof, N. Sastry, D. Wagner  
University of California, Berkeley  
Presented at SenSys 2004

# Goals of TinySec

- Access Control
  - Authorized participants only
- Integrity
  - Altering and retransmitting a message should be difficult
- Confidentiality
- Transparent to applications and programmers

# TinySec Architectural Features

- Single shared global cryptographic key
- Link layer encryption and integrity protection → transparent to applications
  - New radio stack based on original
- Cryptography based on a block cipher
- Two operation modes
  - Authentication
    - Entire packet (data + header) is authenticated with a MAC
  - Authentication + Encryption
    - Payload is encrypted and then MAC is calculated as above

# Block Ciphers

- Keyed pseudorandom permutation
  - DES, RC5, Skipjack, AES
  - Maps  $n$  bits of plaintext to  $n$  bits of ciphertext
- Used to build encryption schemes and message authentication codes (MAC)

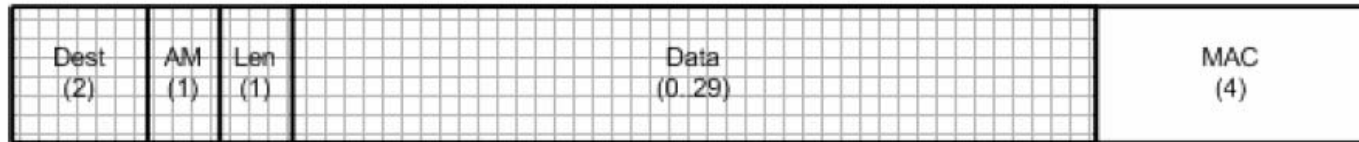
$$E_k : \{0,1\}^n \xrightarrow{K=\{0,1\}^k} \{0,1\}^n$$

- Encryption algorithm used is SkipJack
- IV is set to dst+AM+l+src+counter
  - Counter is 2 bytes long
  - Counter must be persistent across reboot
  - Gives each sender ~65000 messages before IV is reused (worst case)

# Packet Format



(a) TinySec-AE packet format



(b) TinySec-Auth packet format



(c) TinyOS packet format

# Security Analysis

- Access control and integrity
  - Probability of blind MAC forgery  $1/(2^{32})$
  - Replay protection not provided, but can be done better at higher layers
- Confidentiality – Reused IVs can leak information
  - IV reuse will occur after  $2^{16}$  messages from each node [1 msg / min for 45 days]
  - Solutions
    - increase IV length → adds packet overhead
    - key update protocol → adds complexity
  - Applications have different confidentiality requirements
    - Need a mechanism to easily quantify and configure confidentiality guarantees
  - Applications may provide IVs implicitly
    - Apps may be able to guarantee sufficient variability in their messages (eg through timestamps)

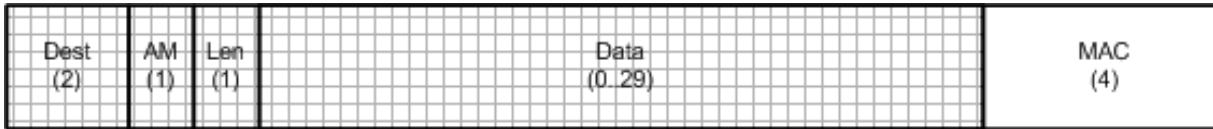
# TinySec Performance

- Characterize Overhead: Energy, Latency, Bandwidth.
- Factors for TinySec overhead
  - Computation
  - Larger Packet Sizes
- Can predict overhead caused by packet sizes
- Measurement goal: Show computation overhead is minimal
- Note: crypto HW only reduces computation overhead

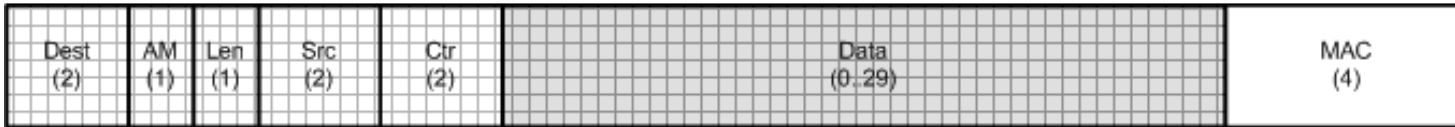
# Packets & Predicted Overhead



Old packet  
(CRC): +7 b



Authentication Only  
(TinySec-Auth): +8 b

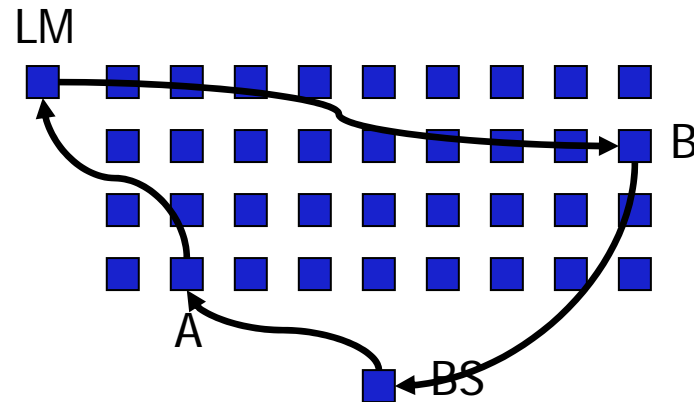


Authentication, Encryption  
(TinySec-AE) : +12 b



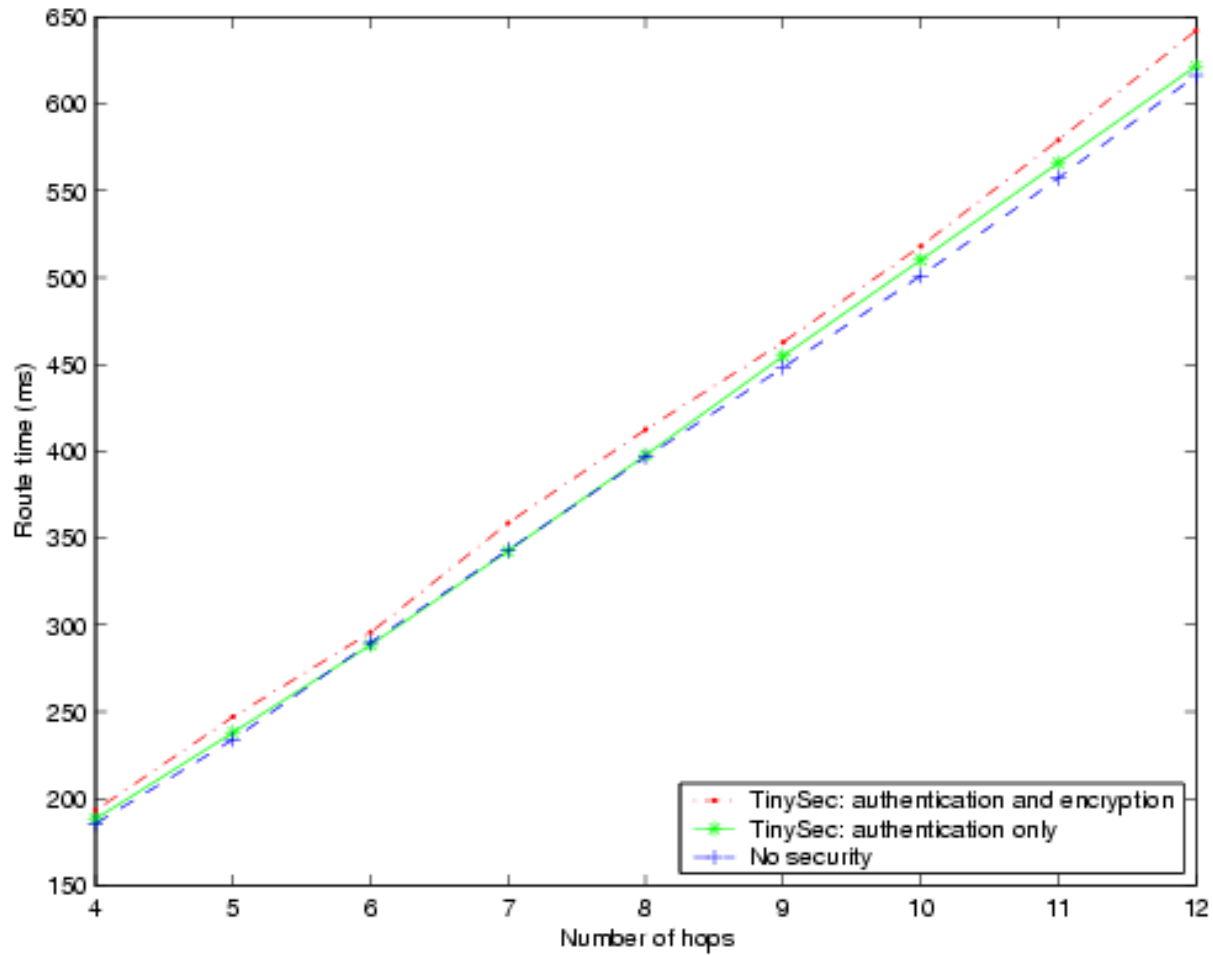
	Overhead (b)	Total Size (b)	Xmit time (ms)	Increase
CRC	39	63	26.2	--
TinySec-Auth	40	64	26.6	1.5%
TinySec-AE	44	68	28.8	8%

# Latency Test Setup

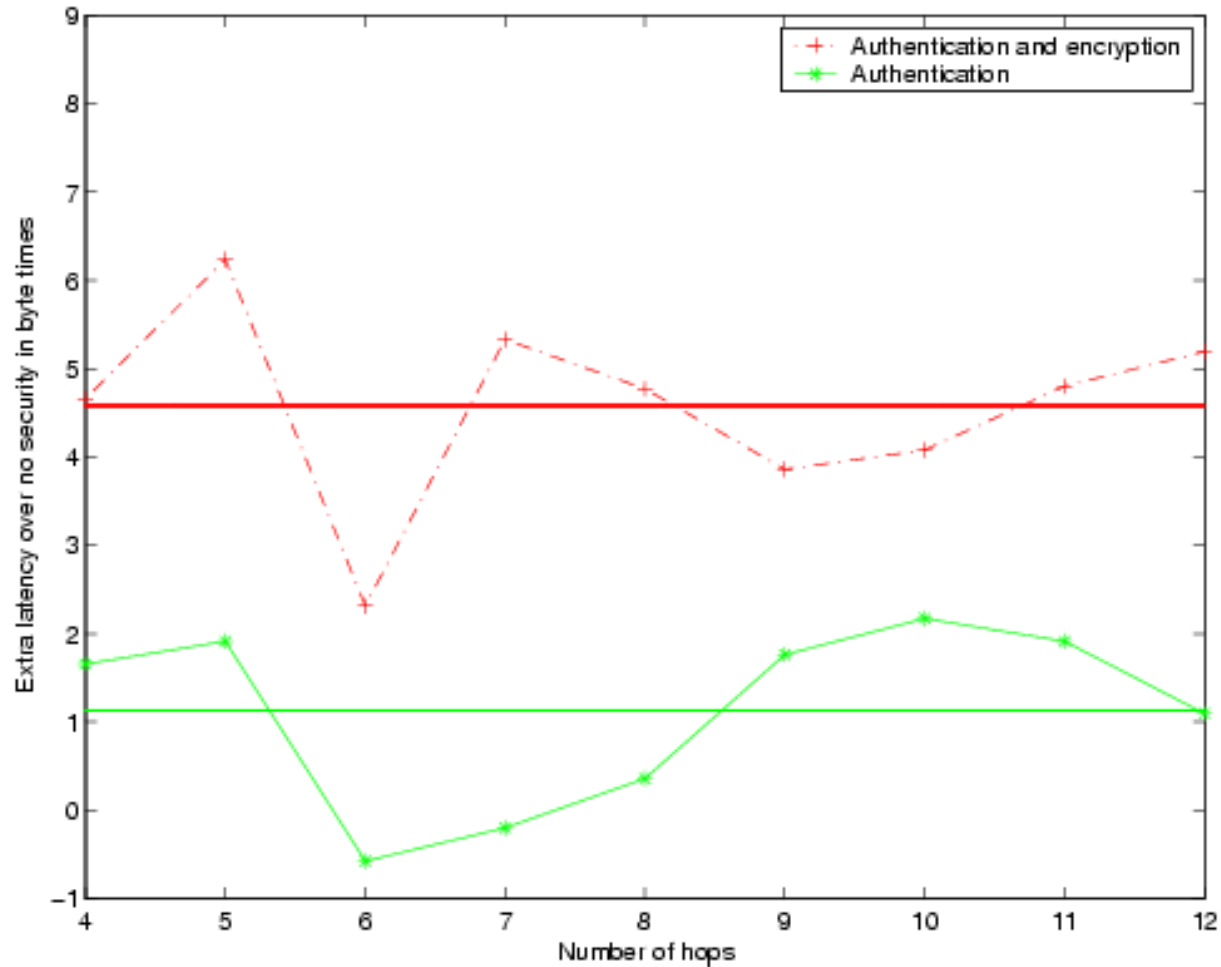


- Setup:
  - 4x9 grid of Mica2s
  - 200 measurements per hopcount
- Test purpose:
  - Measure latency at different hopcounts

# Latency



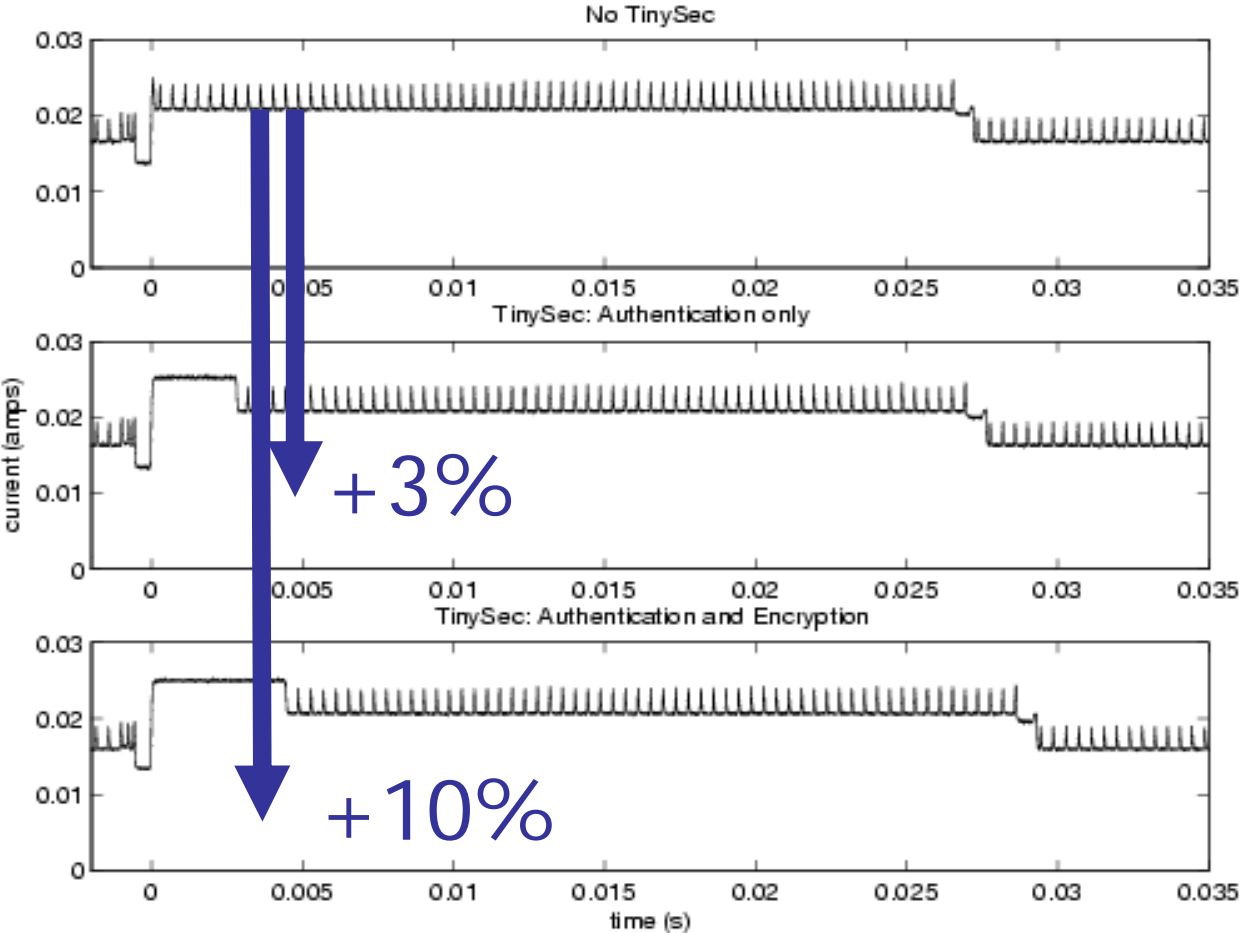
# Latency: Byte Times



# Energy Test Setup

- Single mote transmitting a packet
- Measure voltage drop with oscilloscope

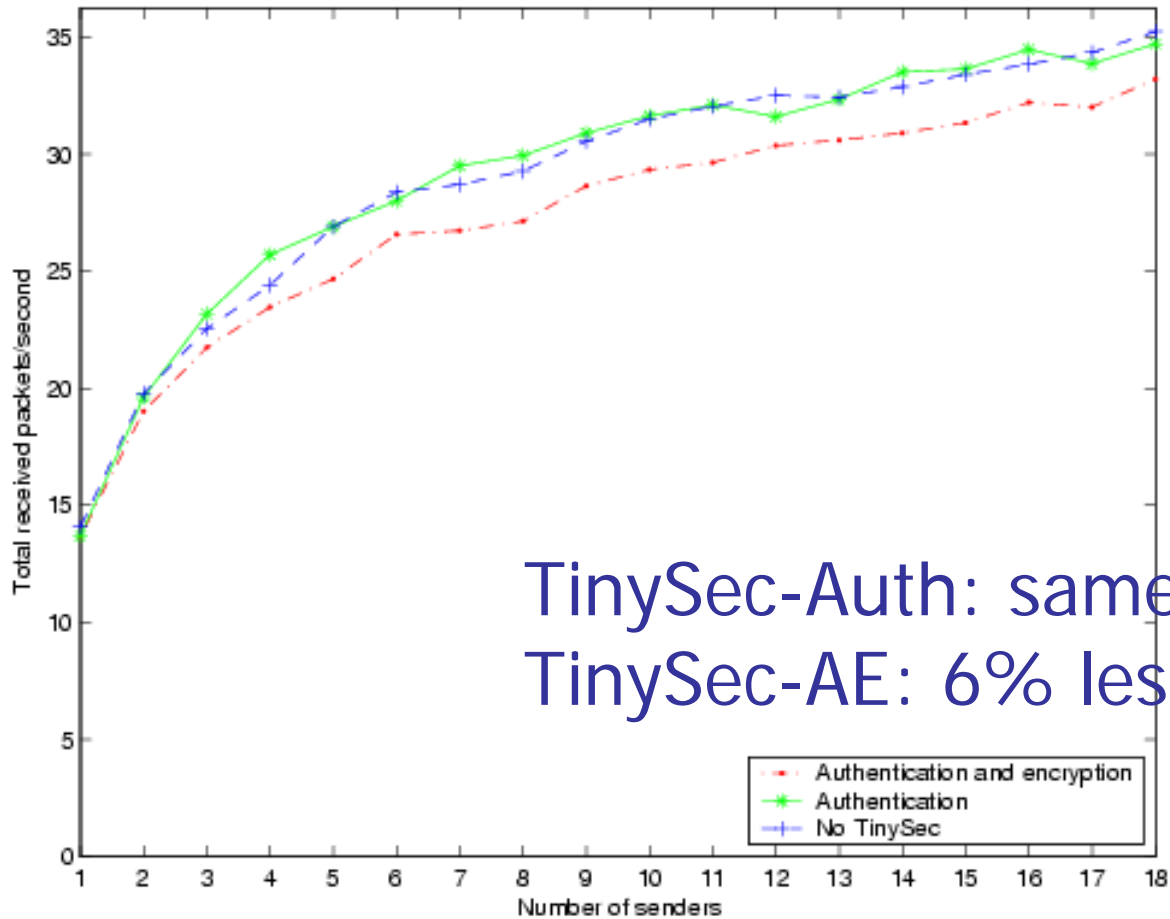
# Energy Consumption



# Bandwidth Test Setup

- Vary number of senders
- Each sender sends as fast as it can
- Measure number of packets successfully received in a time period

# Bandwidth Reduction



TinySec-Auth: same throughput  
TinySec-AE: 6% less throughput

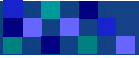
# 802.15.4

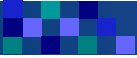


- New standard supported by ChipCon 2240.
- Link-layer security provisions
  - Key management left to higher protocols (ZigBee)
- Design similarities to TinySec:
  - 3 security modes: off, auth, auth + encryption (also include encryption only).
  - Block cipher based
  - 16 byte IV; format similar to TinySec format

## 802.15.4: (cont)

- Design differences to TinySec
  - Larger security parameter choices
    - Performance hit?
  - AES in hardware
  - MAC size variable, 0..16 bytes
  - Encryption: CTR mode
  - Encryption: 16 byte IV. Similar to TinySec Format





- Link-layer security is completely ineffective in the presence of insider attacks or compromised nodes

# Preventing the Sybil Attack

- Using a global key attacker can masquerade as *any* node
  - Could be solved with public key crypto but not feasible due to hardware limitations
- Solution
  - Each node shares unique symmetric key with trusted base station
  - Two nodes use Needham-Schroeder to verify each other's identity and establish shared key
  - Shared key can be used to implement link-layer authentication, encryption
  - Base station can limit the number of neighbors a node can have

# Preventing HELLO Flood attacks

- Verify bi-directionality of the link
  - If attacker also has sensitive radio then node transmission will be heard and ACK will be sent
  - Node should try to authenticate their neighbors
  - Flooding node will have a large number of neighbors, can be detected at the base station

# Wormhole and Sinkhole attacks

- Wormhole and sinkhole attacks are very difficult to defend against
  - Sinkholes are difficult to defend against since it's hard to verify information advertised by nodes
  - Hop-count can be misrepresented through wormhole
- Geographic routing protocols are more resistant to these attacks
  - Wormholes are detected because "neighboring" nodes are outside of radio communication range