Wireless Sensor Network Security

Introduction to wireless sensor networks;

Key establishment;

Node replication attack and detection;

Sensor System Types – Smart-Dust/Motes

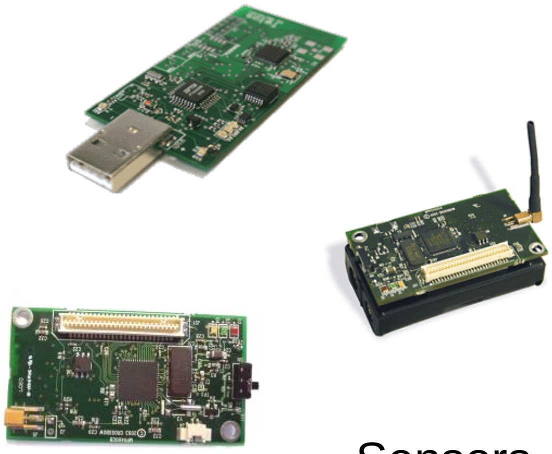
- First introduced in late 90’s by groups at UCB/UCLA/UMich
  - Published at Mobicom/SOSP conferences
  - An integrated computing, communication and sensing platform consisting of millimeter-scale sensor nodes
  - Small enough to remain suspended in air, buoyed by air currents, capable of sensing and communication for hours or days

- Small, resource limited devices
  - CPU, disk, power, bandwidth, etc.
  - Different from vehicular sensing platform where nodes are not energy-starved

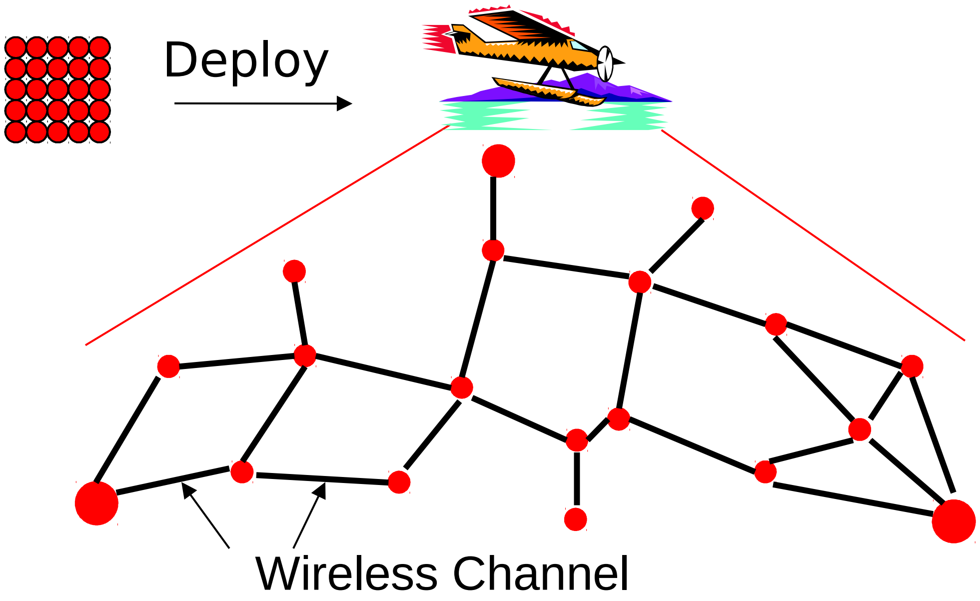
  - Since then, progress in WSN research has yielded major advances toward the original WSN vision

Wireless Sensor Networks (WSNs)

- Consist of a large number of small, cheap, and resource-constrained sensors
- Can be easily deployed in large scale to sense various physical environments

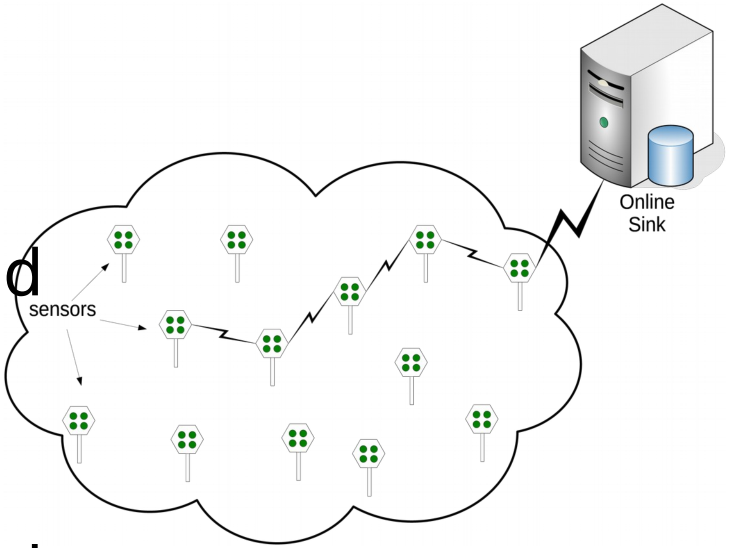


Sensors



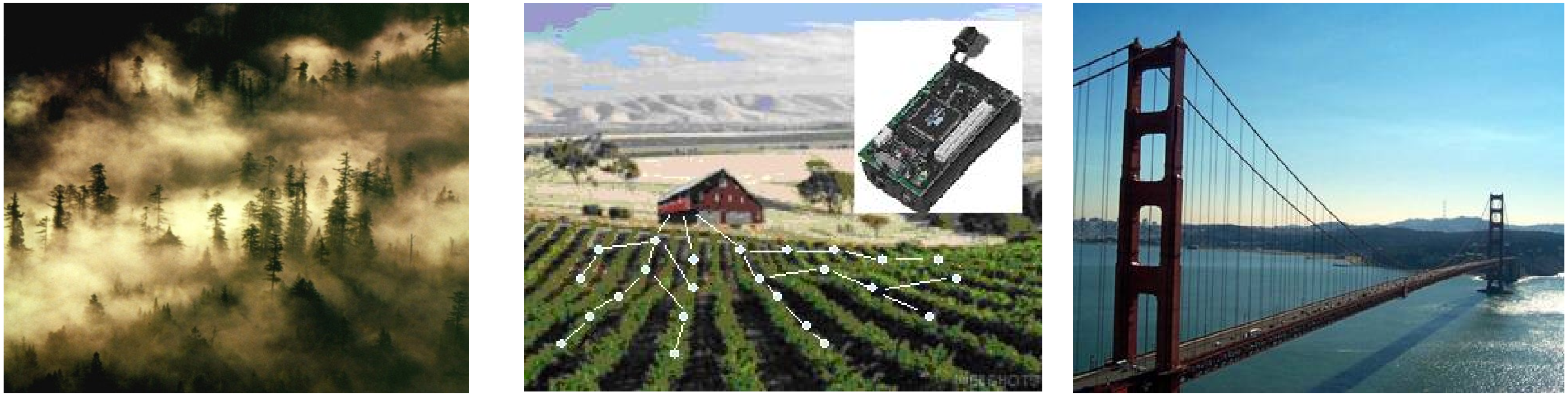
“Typical” Wireless Sensor Networks

- Networking
  - Sensor-to-sink communication (opt. sink-to-sensors)
- Data sensing method
  - Periodic sensing
  - Event driven
  - Query based = on-demand
- On-line sink
  - Real-time off-loading of data



Application Areas

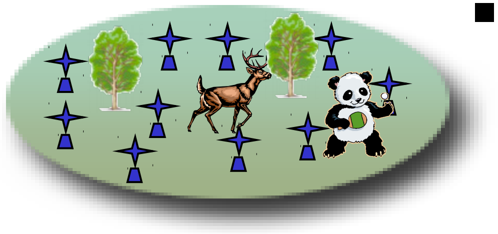
- Military and homeland security
- Industrial sensing, Traffic control
- Environment & Habitat monitoring
- …



Example Application: Parking Space Finder

- A distributed database maintains
  - Spot availability data
  - Address of parking spot
  - Meter description
  - Historical availability data
- Query: Where is the cheapest empty parking spot near Great Hall?
  - Returns list of spaces, details on their meters

More Example Applications



- Anti-poaching WSN in a national park tracking/recording firearm discharge locations

- WSN along an international border monitoring sound and vibration produced by illegal border crossings

Security Requirements in WSNs

- Security is critical to the success of WSN applications!
- Major security requirements in WSNs:
  - Authenticity
    - Enable a sensor to make sure the identities of its communicating parties
  - Integrity
    - Ensures a message being transferred is not corrupted
  - Availability
    - Ensures the survivability of network services
    - Can happen at any layer of sensor networks
  - Confidentiality
    - Ensures data secrecy

Security Challenges in WSNs

- Resource & network constraints:
  - Energy, memory, communication, computation, non-tamper resistant,…
    - Limited energy (battery-powered)
    - Limited computation (4MHz 8-bit)
    - Limited memory (512 bytes)
    - Limited code size(8 Kbytes)
    - Limited communication(30 byte packets)
    - Energy consuming communication
  - Non-tamper resistant,…
  - Wireless medium, infrastructureless, large scale,…
- Major challenges for security design:
  - Efficiency, lightweight, scalability, DoS resilience,…
  - Balance among these competing and even conflicting requirements

Security Research Efforts so far focus on:

- A flurry of research results appeared in early 2000-s addressing a number of WSN security issues:
  - Key management, secure routing, DoS attacks, clone attacks, …
- Solving security problems not specific to WSNs
  - Aiming at miniaturizations of security functionalities (e.g., SPINS, topic today)
- Solving security problems unique to WSNs
  - Clone detection (topic today)
  - Secure aggregation
  - Secure statistical sampling

SPINS: Security Protocols for Sensor Networks

Authors:

- Adrian Perrig,
- Robert Szewczyk
- Victor Wen
- David Culler
- J.D.Tygar

Security Goals

- Data Authentication
- Data Confidentiality
- Data Integrity
- Data Freshness
  - Weak Freshness
    - Partial message ordering, no delay information
    - Useful for sensor measurements
  - Strong Freshness
    - Total ordering on req-res pair, delay estimation
    - Useful for time synchronization

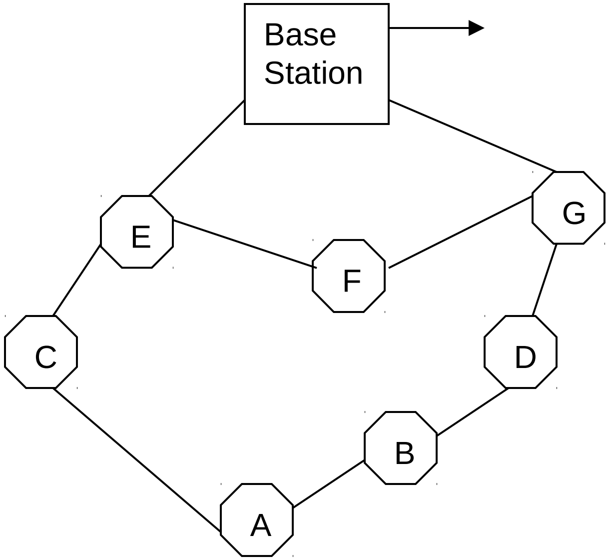
Building Blocks

- SNEP
  - Sensor Network Encryption Protocol
  - Secures point-to-point communication

- μTESLA
  - Micro Timed Efficient Stream Loss-tolerant Authentication
  - Provides broadcast authentication

System Assumptions

- Communication patterns
  -Node to base station (e.g. sensor readings)
  -Base station to node (e.g. specific requests)
  -Base station to all nodes
- Base Station
  -Sufficient memory, power
  -Shares secret key with each node
- Node
  -Limited resources, limited trust



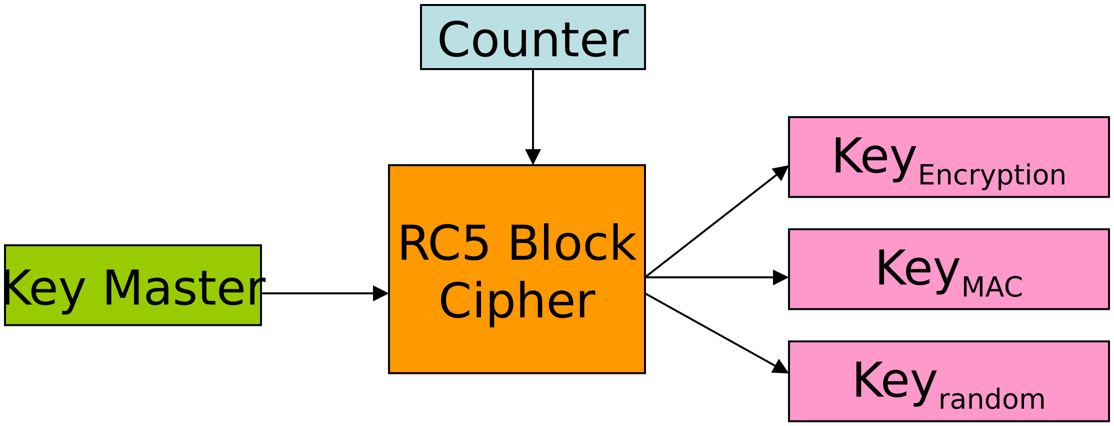
Notation

A, B	Principals(nodes)
N_A	Nonce generated by A
C_A	Counter generated by A
χ_{AB}	Master secret key between A and B (no direction information)
K_{AB}	Secret encryption key between A and B (depends on direction)
K'_{AB}	Secret MAC key between A and B (depends on direction)
$\{M\}_{KAB}$	Encryption of message M with K_{AB}
$\{M\}_{<KAB,IV>}$	Encryption of message M using key KAB and initialization vector IV
$MAC(K'_{AB},M)$	Message authentication code (MAC) of M

SNEP

- Data Confidentiality (Semantic Security)
- Data Authentication
- Replay Protection
- Weak Freshness
- Low Communication Overhead

Key Generation /Setup



Counter

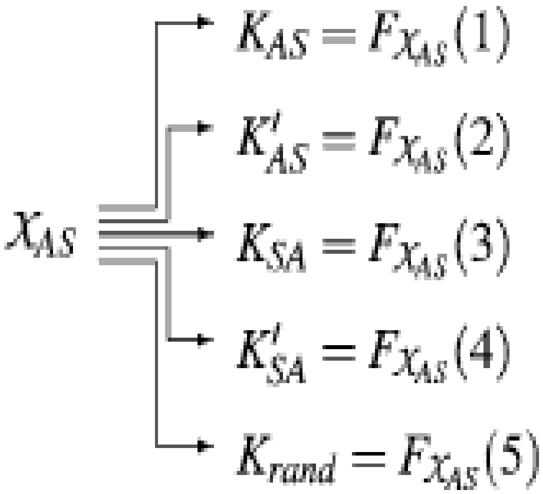
RC5 Block Cipher

Key Master

$\text{Key}_{\text{Encryption}}$

Key_{MAC}

$\text{Key}_{\text{random}}$



χ_{AS}

$K_{AS} = F_{\chi_{AS}}(1)$

$K'_{AS} = F_{\chi_{AS}}(2)$

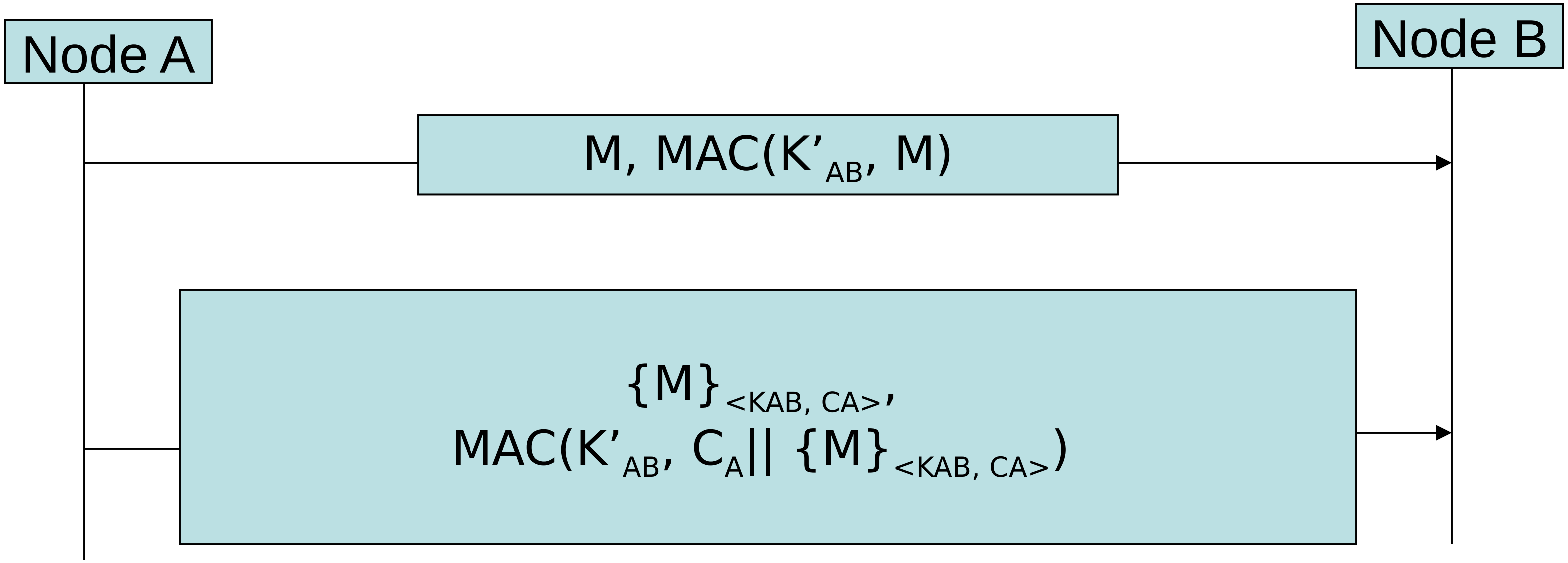
$K_{SA} = F_{\chi_{AS}}(3)$

$K'_{SA} = F_{\chi_{AS}}(4)$

$K_{rand} = F_{\chi_{AS}}(5)$

- Nodes and base station share a master key pre-deployment
- Other keys are bootstrapped from the master key:
  - Encryption key
  - Message Authentication code key
  - Random number generator key

Authentication, Confidentiality



Node A

Node B

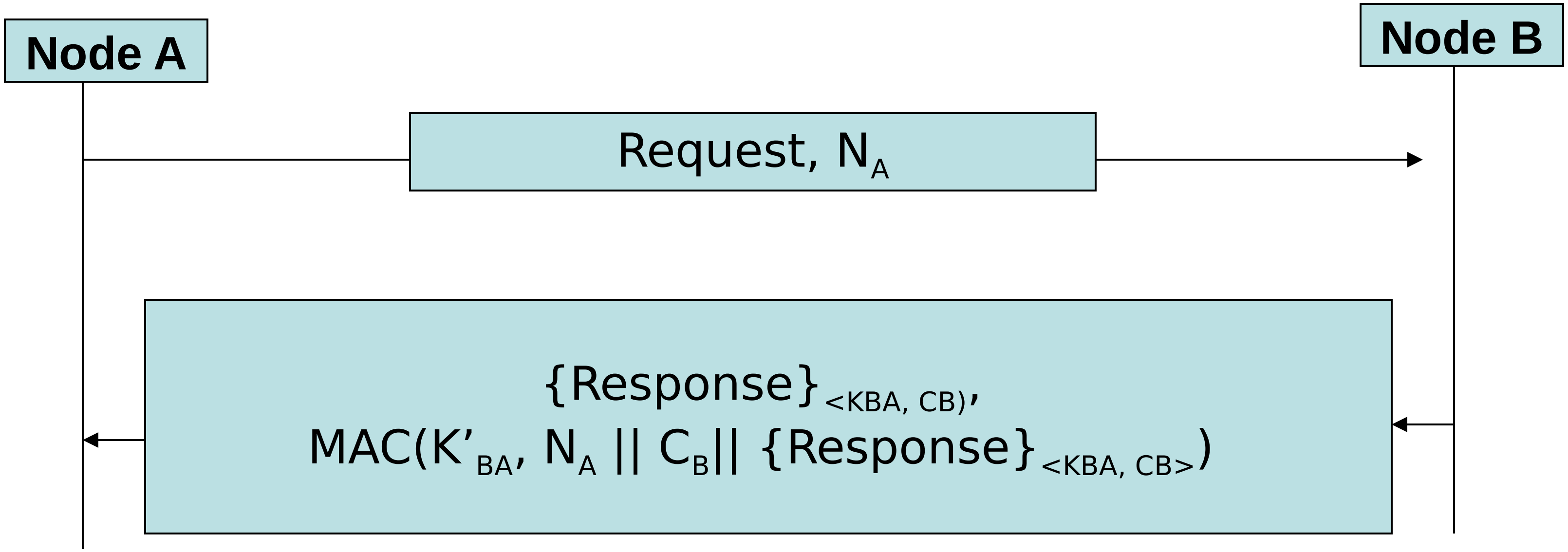
$M, MAC(K'_{AB}, M)$

$\{M\}_{<KAB, CA>}$,

$MAC(K'_{AB}, C_A || \{M\}_{<KAB, CA>})$

- Without encryption can have only authentication
- For encrypted messages, the counter is included in the MAC
- Base station keeps current counter for every node

Strong Freshness



Node A

Node B

Request, N_A

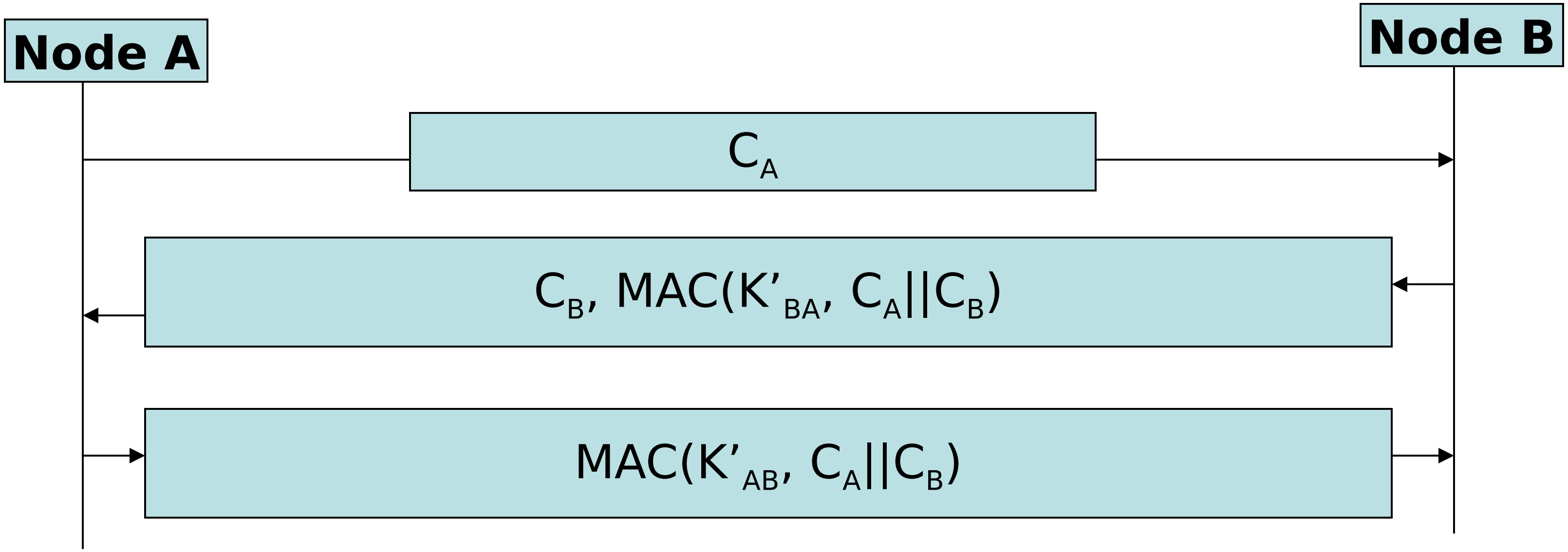
$\{\text{Response}\}_{<KBA,\ CB)}$,

$MAC(K'_{BA}, N_A \mid\mid C_B \mid\mid \{\text{Response}\}_{<KBA,\ CB>})$

- Nonce generated randomly
- Sender includes Nonce with request
- Responder include nonce in MAC, but not in reply

Counter Exchange Protocol

- Bootstrapping counter values



Node A

Node B

C_A

C_B, MAC(K'$_{BA}$, $C_A || C_B$)

MAC(K'$_{AB}$, $C_A || C_B$)

To synchronize:

$A \rightarrow B : \quad N_A$

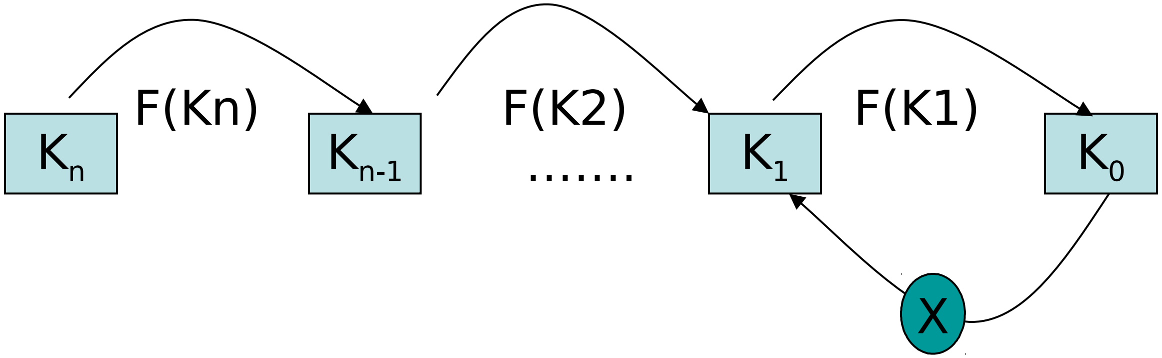
$B \rightarrow A : \quad C_B$, MAC$(K'_{BA}, N_A \,||\, C_B)$.

µTESLA : Authenticated Broadcast

- TESLA : efficient source authentication in multicast for wired networks.

- Problems with TESLA
    - -Digital Signature for initial packet authentication
        - µTESLA uses only symmetric mechanism
    - -Overhead of at least 16 bytes per packet (8-byte MAC and key)
        - µTESLA discloses key once per epoch
    - -One way key chain is too big
        - µTESLA restricts number of authenticated senders

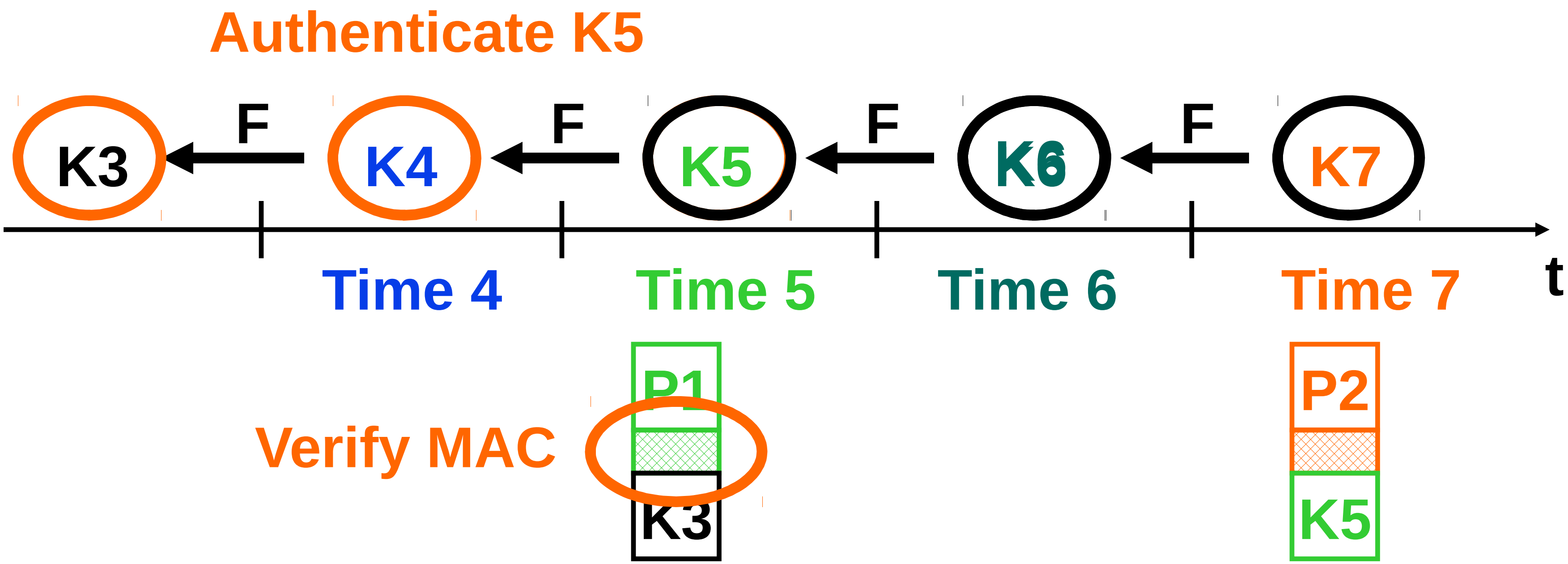
Key Setup



- Main idea: One-way key chains
- K_0 is initial commitment to chain, known by the sensor
- Base station gives K_0 to all nodes

µTESLA Quick Overview I

- Keys disclosed 2 time intervals after use
- Receiver knows authentic K3
- Authentication of P1:MAC(K5,P1)



Authenticate K5

K3 ← F K4 ← F K5 ← F K6 ← F K7

Time 4 Time 5 Time 6 Time 7 t

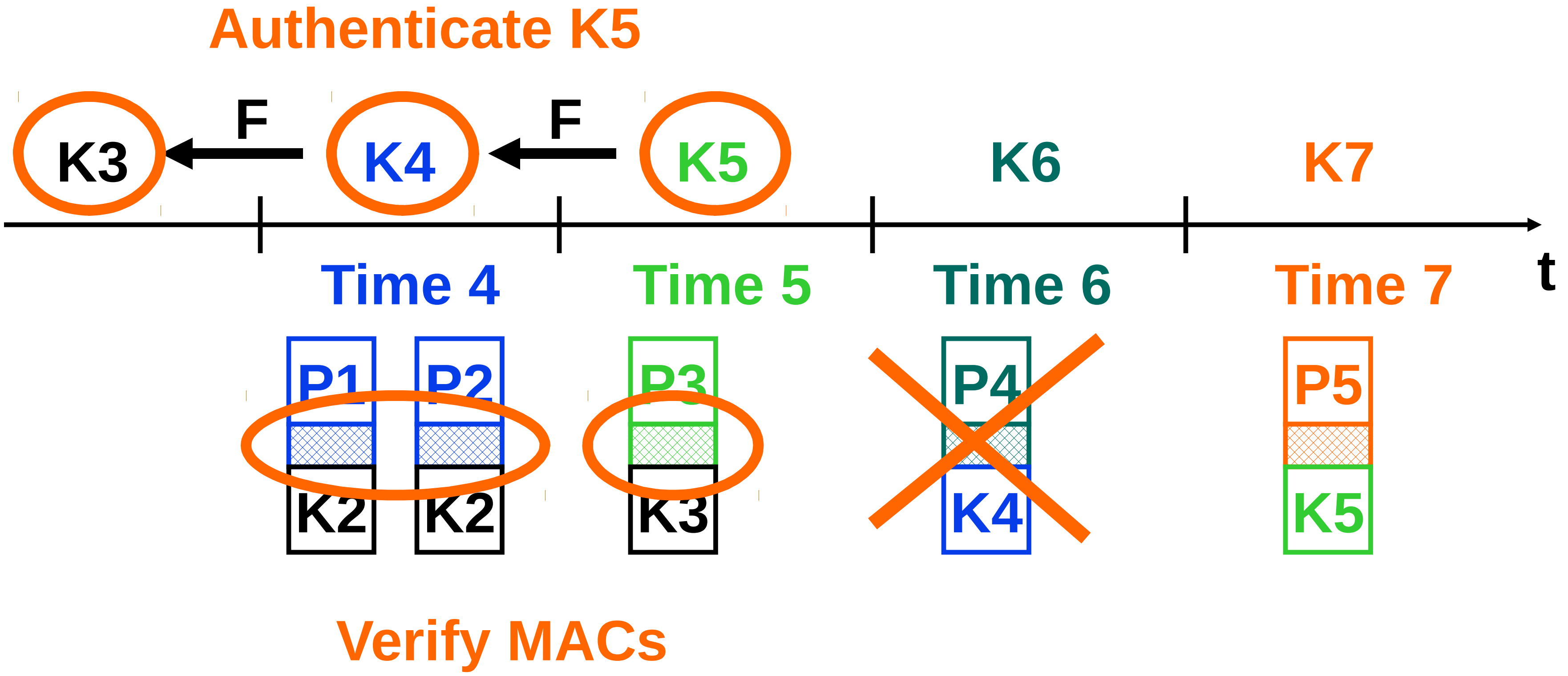
Verify MAC

P1 | K3

P2 | K5

μTESLA Quick Overview II

- Perfect robustness to packet loss



μTESLA Properties

- Asymmetry from delayed key disclosure
- Self-authenticating keys
- Requires loose time synchronization
- Low overhead (1 MAC)
    - Communication (same as SNEP)
    - Computation (~ 2 MAC computations)
- Independent of number of receivers

Applications built from SPINS

- Authenticated Routing
- Node to Node Key Agreement (using base station as the trusted party)

$A \rightarrow B$: N_A, A

$B \rightarrow S$: $N_A, N_B, A, B, MAC(K'_{BS}, N_A \,||\, N_B \,||\, A \,||\, B)$

$S \rightarrow A$: $\{SK_{AB}\}_{KSA}$, $MAC(K'_{SA}, N_A \,||\, A \,||\, \{SK_{AB}\}K_{SA})$

$S \rightarrow B$: $\{SK_{AB}\}_{KSB}$, $MAC(K'_{SB}, N_B \,||\, B \,||\, \{SK_{AB}\}K_{SB})$

SPINS Conclusion

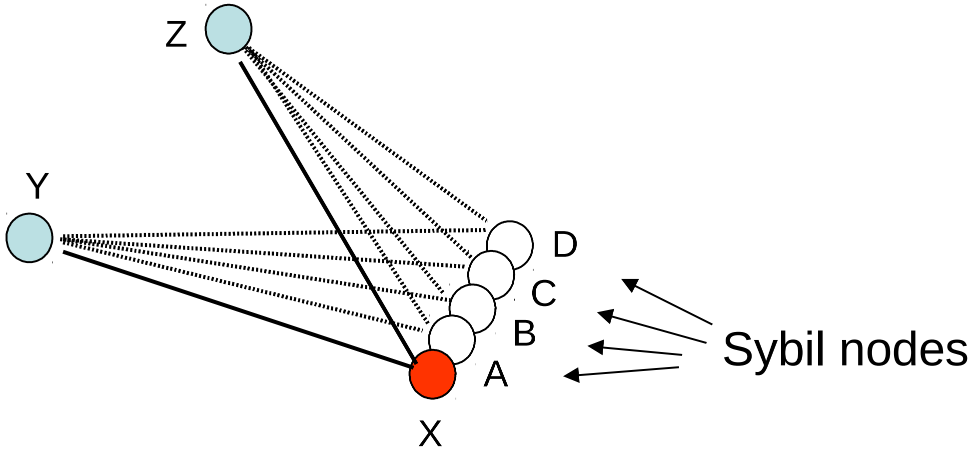
Advantages

- Strong security protocols affordable
    - First broadcast authentication
- Low security overhead
    - Computation, memory, communication
- Apply to future sensor networks
    - Energy limitations persist
    - Tendency to use minimal hardware
- Base protocol for more sophisticated security services

Distributed Detection of Node Replication Attacks in Sensor Networks

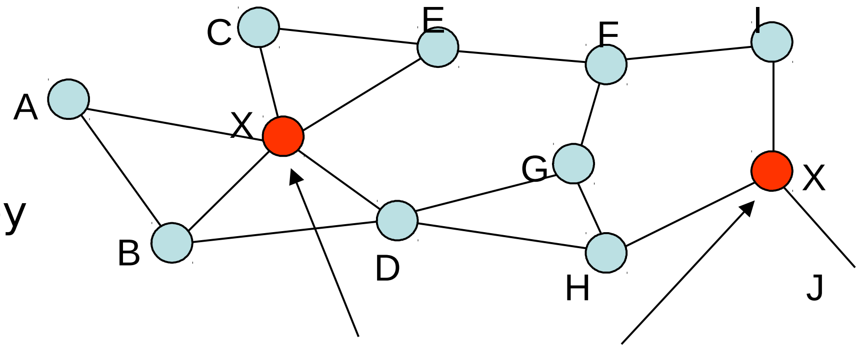
Sybil vs replication attacks

- Sybil Attacks
  - One node has multiple valid identifications



Sybil nodes

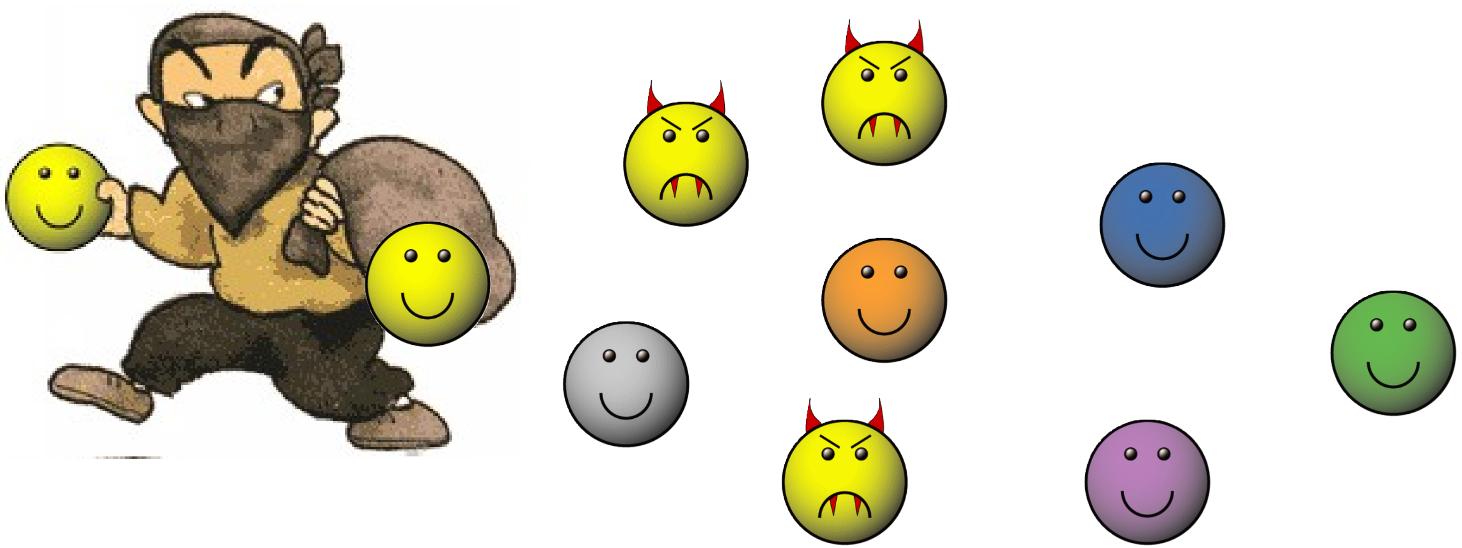
- Replication Attacks
  - Multiple nodes have the same identification
  - Capturing many nodes is hard
  - Instead, capture one node and copy it



replicated nodes

Replication is Easy

- Only need to capture one node
- Offline attack to extract node’s secrets
- Transfer secrets to generic nodes
- Deploy clones



Repercussions

- Clones know everything compromised node knew
- Adversary can …
  - Inject false data or suppress legitimate data
  - Spread blame for abnormal behavior
  - Revoke legitimate nodes using aggregated voting
  - Monitor communication

Detection Approaches

- Centralized Detection

A key-management scheme for distributed sensor networks, by L. Eschenauer, V. Gligor, ACM Conference on Computer and Communication Security (CCS) 2002

- Localized Detection

Random key predistribution schemes for sensor networks, by H. Chan, A. Perrig, D. Song, IEEE Symposium on Security and Privacy 2003

- Distributed Detection

Distributed Detection of Node Replication Attacks in Sensor Networks, by Bryan Parno, Adrian Perrig, Virgil Gligor, IEEE Symposium on Security and Privacy 2005

Centralized Detection

A key-management scheme for distributed sensor networks, by L. Eschenauer, V. Gligor, ACM Conference on Computer and Communication Security (CCS) 2002

- Each node sends neighbor list to a central base station
  - Base station searches lists for duplicates
  - Disadvantages
    - Some applications may not use base stations
    - Single point of failure
    - Exhausts nodes near base station (and makes them attack targets)

Localized Detection

Random key predistribution schemes for sensor networks, by H. Chan, A. Perrig, D. Song, IEEE Symposium on Security and Privacy 2003

- Neighborhoods use local voting protocols to detect replicas
- Disadvantage
  - Replication is a global event that cannot be detected in a purely local fashion

Distributed Detection

Distributed Detection of Node Replication Attacks in Sensor Networks, by Bryan Parno, Adrian Perrig, Virgil Gligor, IEEE Symposium on Security and Privacy 2005

- Goals:
  - Detect replication with high probability
  - After protocol concludes, legitimate nodes have revoked replicas
  - Secure against adaptive adversary
    - Unpredictable to adversary
    - No central points of failure
  - Minimize communication overhead
- Two Preliminary Schemes
  - Node-to-Network Broadcast
  - Deterministic Multicast
- Two Primary Schemes
  - Randomized Multicast
  - Line Select Multicast

Distributed Detection

- Assumptions
  - Public key infrastructure
    - Occasional elliptic curve cryptography is reasonable
    - Can be replaced with symmetric mechanisms
  - Network employs geographic routing
  - Nodes are primarily stationary

Node-to-Network Broadcast (1)

- Each node uses an authenticated broadcast message to flood the network with its location information.
- Each node stores the location information for its neighbors.
- If conflicting claim is detected, the offending node is revoked.

Node-to-Network Broadcast (2)

- Simple and achieve 100% detection rate
- Each node stores location information for its d neighbors.

- Total communication cost is O(n^2)

Deterministic Multicast (1)

- Each node broadcasts its location to its neighbors.
  - Neighbors forward location claim to a subset of the nodes “witnesses”: $F(\alpha) = W_1, W_2, \ldots, W_g$
    - Coupon Collector Problem: each node only needs to select (glng)/d random destinations from the set of witnesses.
  - Once the witness detects a location conflict, it revokes the node by flooding.

Deterministic Multicast (2)

- Average path length is O($\sqrt{n}$), then communication cost is $O(\frac{g \ln g \sqrt{n}}{d})$

- F is a deterministic function, an adversary can also determine all witness nodes.
  - Better security guarantee, larger g -> larger communication cost

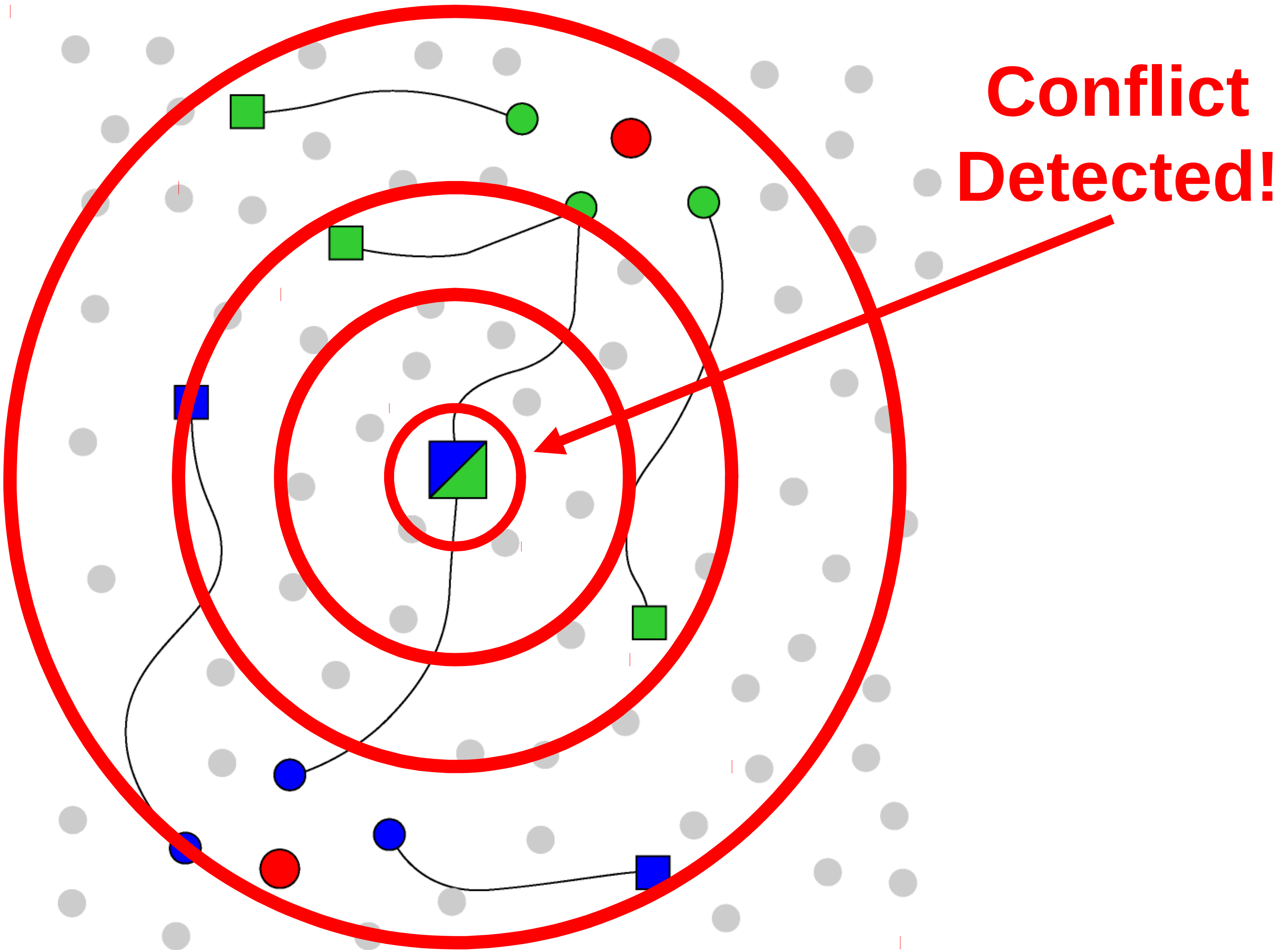
Primary Approaches Overview

- **Step 1: Announce locations**
  - Each node signs and broadcasts its location to neighbors
    - Location = (x,y), virtual coordinates, or neighbor list
  - Nodes must participate or neighbors will blacklist them
- **Step 2: Detect replicas**
  - Location claims are sent to “witness” nodes by neighbors
  - Ensures at least one “witness” node receives two conflicting location claims
- **Step 3: Revoke replicas**
  - Witness floods network with conflicting location claims
  - Signatures prevent spoofing or framing

Randomized Multicast Protocol

- Each node signs and broadcasts its location to neighbors
- Each neighbor forwards location to “witness” nodes
  - Witness chosen at random by selecting random geographic point and forwarding message to node closest to the point
  - Each neighbor selects $\sqrt{n}/d$ witnesses for a total of $\sqrt{n}$ points
- Birthday Paradox implies location claims from a cloned node and its clone will collide with high probability
- Conflicting location claims are evidence for revoking clones
- Signatures prevent forgery of location claims

Randomized Multicast Detection



Conflict Detected!

Randomized Multicast Analysis

- High probability of detection

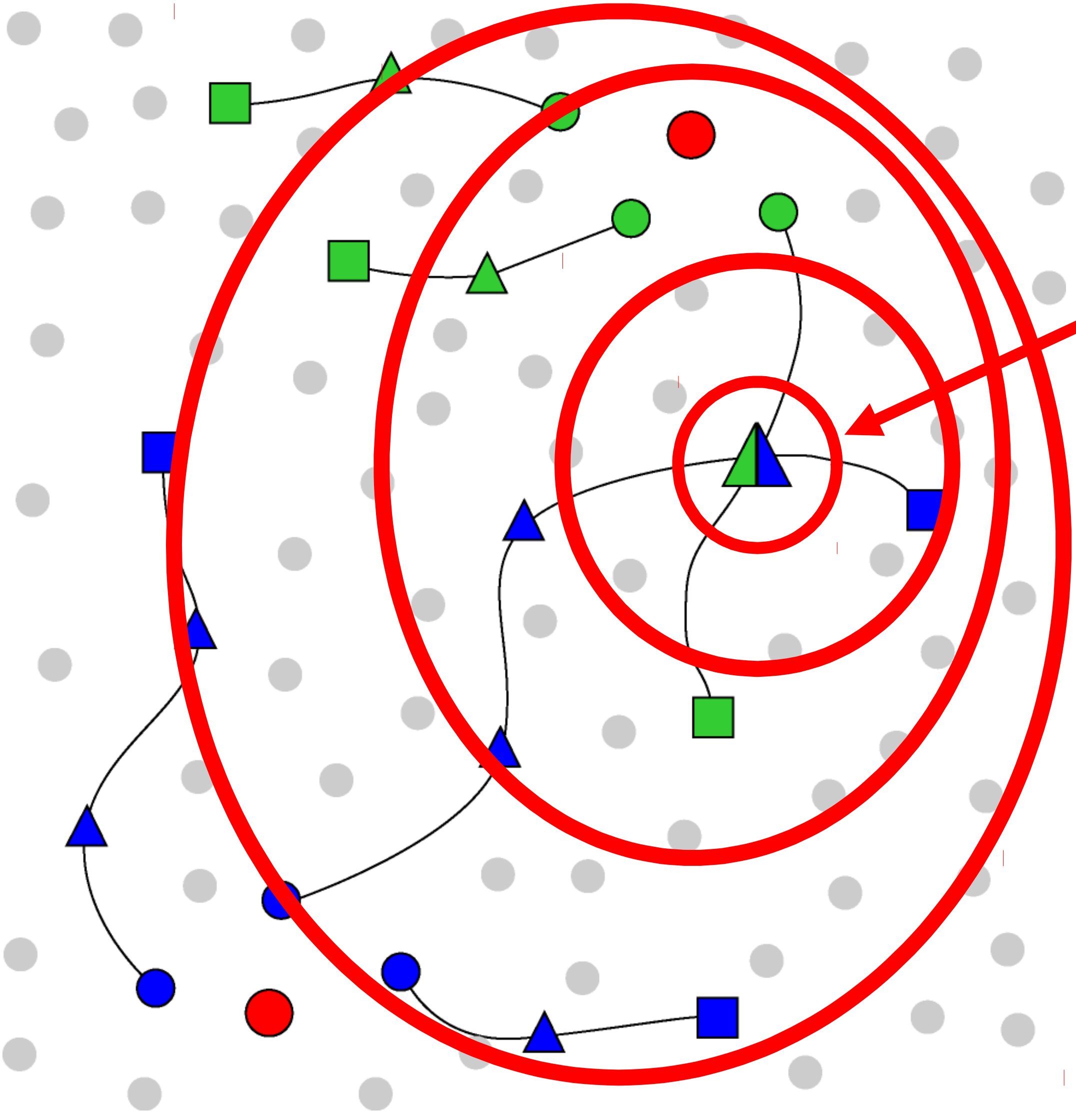
$$P_{Detect} \geq 1 - e^{\frac{-w^2 R}{n}}$$

  - 2 replicas (R=2), w = $\sqrt{n}$, $P_{Detect} \geq 95\%$,
- Decentralized and randomized
- Moderate communication overhead
  - Each node’s location sent to $\sqrt{n}$ witnesses
  - Path between two random points in the network is O($\sqrt{n}$) hops on average
  - Results in O(n) message hops per node
  - Total $O(n^2)$

Line-Selected Multicast Protocol

- In a sensor network, nodes route data as well as collect it
- Again, neighbors forward location claim to “witness” nodes
- Each intermediate node checks for a conflict and forwards the location claim
- If any two “lines” intersect, the conflicting location claims provide evidence for revoking clones

Line-Selected Multicast Detection



Conflict Detected!

Line-Selected Multicast Analysis

- High probability of intersection for two randomly drawn lines in square area
  - Only need a constant number of lines (e.g. for 5 lines/node, $P_{Detect} \geq 95\%$)
- Decentralized and randomized
- Minimal communication
  - Line segments $O(\sqrt{n})$ on average
  - Only requires $O(\sqrt{n})$ message hops per node
  - Total: $O(n^{3/2})$

Conclusion

- Distributed detection solutions seem more reasonable
- Still best communication overhead is $O(n^{3/2})$