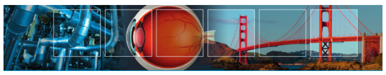
Chapter 11: Security



Chapter 11: Roadmap

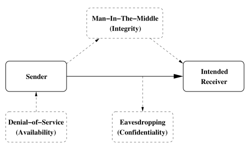
- Security fundamentals
- Security challenges
- Security attacks
- Security protocols and mechanisms
- IEEE 802.15.4 and ZigBee security

Fundamentals

- Security and privacy are big challenges for any type of computing and networking environment
- Well-known CIA security model:
  - Confidentiality
    - ensure that only the intended receiver can read/interpret a message
    - unauthorized access is prevented
  - Integrity
    - ensure that a message cannot be modified
    - unauthorized individuals should not be able to destroy/alter message
  - Availability
    - ensure that system/network is able to perform its tasks without interruption
    - often measured in terms of percentages of up/down time

Examples

- Confidentiality:
  - eavesdropping: unauthorized message reception
- Integrity:
  - man-in-the-middle attack: unauthorized individual/system positions itself between sender and receiver to intercept, modify, and retransmit messages
- Availability:
  - denial-of-service attack: attempt to disrupt transmission or service



More Terminology

- Authentication
  - process of establishing or confirming the identity of user/device
  - ensures that message came from who it claims to have come from
- Nonrepudiation
  - process of proving that person/device has performed a certain transaction/transmission
- Digital signatures
  - often used to support authentication, nonrepudiation, and integrity

Cryptography

- Process of protecting information using encoding/decoding techniques
- Symmetric key cryptography
  - single key shared between communicating parties
  - simple example: shift cipher (key = fixed shift in alphabet)
  - challenge: secure distribution of shared key
  - examples: DES, AES, IDEA
- Public key cryptography
  - secret key: will never be shared with anyone else
  - public key: can be shared freely
  - message encrypted with secret key can only be decrypted with corresponding public key (e.g., for authenticating the sender)
  - message encrypted with public key can only be decrypted with corresponding secret key (e.g., for providing confidentiality)
  - examples: RSA, Diffie-Hellman agreement protocol

Challenges of Security in WSNs

- Resource constraints
  - limited computational, networking, and storage capabilities of sensors
  - energy constraints of sensors
- Lack of central control
  - large WSNs often don't have centralized control
  - requires distributed/decentralized security solutions
- Remote location
  - sensors often left unattended
  - difficult to prevent unauthorized physical access and tampering
- Error-prone communication
  - difficult to distinguish wireless communication errors from attacks

Security in WSNs

- WSN characteristics that facilitate security:
  - self-managing and self-repairing nature
  - redundancy
- Data freshness problem
  - WSN security must ensure that sensor data are recent (and not replays of old data)
  - particularly important for key distribution schemes
- WSNs provide more opportunities for attacks than other networks
  - many sensor protocols require location information
  - many sensor nodes require accurate time synchronization
  - both can be affected by modifying, injecting, dropping messages (e.g., beacons) carrying such information

Denial-of-Service (DoS)

- Attempt to stop network/system from functioning or providing a service
- Physical Layer DoS
  - jamming attack
    - interfere with the radio frequencies of a WSN
    - even small numbers of attacking nodes can be effective if well positioned (e.g., close to an important node such as a BS) or if their signals are strong
    - countermeasure: spread-spectrum communication (e.g., FHSS)
  - tampering attack
    - attacker obtains physical access to sensor node
    - used to modify/destroy node, obtain sensitive information or use as entry points for further attacks into the network
    - countermeasures: tamper-proof materials and enclosures, disable device when attack detected

Denial-of-Service (DoS)

- Link Layer DoS
  - collision attack
    - attempt to interfere with packet transmissions
    - causes costly exponential backoff procedures and retransmissions
    - often tries to cause collisions near the end of a frame, requiring retransmission of entire frame
  - exhaustion attack
    - attacks (such as collision attack) with the goal of premature depletion of a sensor's energy sources
    - example: issue RTS message to prompt CTS response from another node (exploiting handshake techniques)

Attacks on Routing

- Blackhole attack
  - malicious node on a route simply drops all packets
- Selective forwarding attack
  - similar to blackhole attack, but not all traffic is dropped
  - more difficult to detect (hard to distinguish attack from poor connectivity)
- Rushing attack
  - exploits route discovery techniques of on-demand protocols
  - route request packets are rushed towards destination, increasing the malicious node's probability to be on the selected route
- Sinkhole attack
  - node attempts to position itself on as many network flows as possible
- Sybil attack
  - attacker claims to have multiple identities or locations
- Wormhole attack
  - out-of-band (bandwidth-rich) connection between attackers used to face short path to the gateway, attracting many flows to these nodes

Attacks on Transport Layer

- Flooding attack
  - exploits fact that many transport protocols maintain state information and are therefore vulnerable to memory exhaustion
  - example: attacker makes many (incomplete) connection requests, forcing a node to allocate more and more resources
- Desynchronization attack
  - attempt to disrupt communication between nodes by repeatedly forging messages to these nodes
  - example: fake packets carry old sequence numbers to make a node believe that its previous transmissions were not correctly received

Attacks on Data Aggregation

- Aggregation (and fusion) operations are often easily affected by an attacker
  - average function $f(x_1,\ldots,x_n)=(x_1+\ldots+x_n)/n$
    - replacing a single measurement x_1 with a fake reading x_1^*, the average will change from $y=f(x_1,\ldots,x_n)$ to $f(x_1^*,x_2,\ldots,x_n) = y+(x_1^*-x_1/n)$
    - attacker can choose x_1^* and thereby determine outcome of aggregation
  - sum function $f(x_1,\ldots,x_n)=x_1+\ldots+x_n$
    - replacing a single measurement x_1 with a fake reading x_1^*
  - minimum function $f(x_1,\ldots,x_n)=\min(x_1,\ldots,x_n)$
    - replacing a single reading does not always lead to incorrect aggregation
    - replacing x_1 with x_1^* raises minimum if x_1 is unique smallest reading of all x_i
    - replacing any xi with very small value can lower the minimum
    - similarly true for maximum function
  - count function: each sensor contributes 0 or 1 to the result
    - changing k readings changes result by at most k
    - may be negligible if k is small compared to the number of measurements

Privacy Attacks

- Attempts to obtain sensitive information collected and communicated in WSNs
- Eavesdropping
  - made easy by broadcast nature of wireless networks
- Traffic analysis
  - used to identify sensor nodes of interest (data of interest), sensor nodes that are vulnerable, and sensor nodes that are critical to the correct operation of the entire network (e.g., gateway devices, cluster heads)

Symmetric versus Public Key

- Public key cryptography
  - used to provide confidentiality, integrity, and authentication
  - computationally expensive
  - some implementations for resource-constrained devices exist (ECC, elliptic curve cryptography)
- Symmetric key cryptography
  - more resource-efficient
  - problem of key distribution

Key Management

- Reliable and secure establishment of shared cryptographic keys
- Peer Intermediaries for Key Establishment (PIKE)
  - uses sensor nodes as trusted intermediaries for key distribution
  - every sensor shares different pairwise key with each of O(√n) nodes
  - for any pair of nodes A and B, there is at least one node, C, that shares a pairwise key with both A and B
  - each sensor has an ID *(x,y)* and the network is represented as a matrix with √n rows and columns, where a node's position in the matrix it its node ID
  - then, each node *(x,y)* shares a pairwise key with each node in the following two sets:

$$(i,y)\forall i \in \{0,1,2,...,\sqrt{n}-1\}$$
$$(x,j)\forall j \in \{0,1,2,...,\sqrt{n}-1\}$$

Key Management

- PIKE (contd.)
  - example: node *(x,y)* will share key $K_{(x,y),(1,y)}$ with node *(1,y)* and another key $K_{(x,y)(2,y)}$ with node *(2,y)*
  - a node will maintain 2(√n-1) keys

00	01	02	03	04	05	...	09
10	11	12	13	14	15	...	19
20	21	22	23	24	25	...	29
30	31	31	33	34	35	...	39
.	.	.	.	.	.		.
.	.	.	.	.	.		.
.	.	.	.	.	.		.
90	91	92	93	94	95	...	99

Defenses Against DoS Attacks

- Jamming attacks
  - isolate affected region by re-routing traffic
  - use spread-spectrum techniques
- Collision and exhaustion attacks
  - error-correcting codes
  - rate-limiting schemes
- Spoofing and alteration
  - message authentication codes (MAC)
- Path-based DoS
  - attacker overwhelms nodes by flooding a multi-hop end-to-end communication path with replayed or injected packets
  - one-way hash chains can be used to validate received packets

Defenses Against Aggregation Attacks

- Delayed aggregation and delayed authentication
  - base station generates a one-way key chain using a public one-way function F, where $K_i=F(K_{i+1})$
  - each device stores key K_0 before deployment ($K_0=F^n(K)$, i.e., F applied to a secret key n times)
  - first base station transmissions are encrypted using $K_1=F^{n-1}(K)$
  - once all messages transmitted using K_1 have been received:
    - the base station reveals K_1
    - all nodes compute $F(K_1)=F(F^{n-1}(K))$ and verify that it matches $K_0=F^n(K)$
    - sensor nodes decrypt the messages

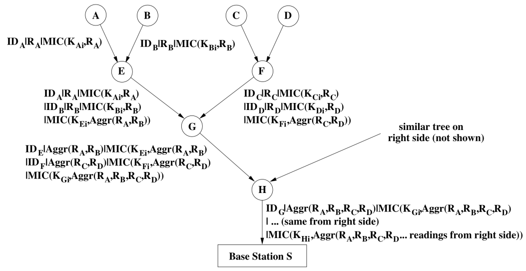
Defenses Against Aggregation Attacks

- Example:
  - nodes A-D send messages to the base station, each node's message contains the sender's ID, the sensor data, and a MAC calculated over the data using a temporary key
  - parent node cannot yet verify the MAC since it doesn't have the child's key
  - parent node stores this message and retransmits it to its own parent after certain timeout value
  - E's message to parent G contains messages received from its children (nodes A and B) and a MAC computed over the aggregate of A's and B's data using E's key
  - this process continues, i.e., every parent combines data from its children and adds its own MAC over the aggregate using its own key
  - once the base station receives messages from its children, it can compute the final aggregate value

Defenses Against Aggregation Attacks

- Example (contd.):
  - base station has shared temporary key with each sensor, therefore it can verify whether a received message was transmitted by H by calculating the MAC of the aggregation using K_{Hi} and comparing it to the MAC in the message
  - this validates that H sent the final message, but it does not validate that the message correctly reflects the readings from the other nodes
  - base station reveals the temporary keys to the network by sending each key (along with a MAC) to all sensor nodes using its own current key K_i
  - base station sends out its current key K_i such that nodes can check the MAC values and to advance to the next key in the chain

Defenses Against Aggregation Attacks



Defenses Against Routing Attacks

- Attacks from "outside" versus "inside" the network
  - link-layer encryption and authentication can prevent adversary from joining a network, preventing many possible routing attacks
  - these techniques may be ineffective if network is attacked from the inside (e.g., using a compromised node)
- Sybil attacks
  - sensor nodes can share unique symmetric key with a trusted base station to verify each other's identity
  - base station can also limit the number of neighbors a node is allowed to have (i.e., a compromised node can communicate with only a few other nodes)
- Sinkhole attacks
  - difficult to defend against where protocols are used that establish routes based on information that it difficult to verify (e.g., energy)
  - easier for routes based on minimum hop counts, but hop counts can be misrepresented through a wormhole
  - with geographic routing, it is difficult to redirect traffic elsewhere to create a sinkhole

Defenses Against Routing Attacks

- Rushing attacks
  - secure neighbor detection approach can be used to allow sender and receiver of a route request to verify that the other party is in fact within normal transmission range
  - example of a three-round mutual authentication protocol:
    - sender sends a neighbor solicitation packet
    - receiver responds with neighbor reply packet
    - sender sends a neighbor verification message (which includes broadcast authentication of a timestamp and the link from the source to the destination)

Security Protocols for Sensor Networks

- SPINS provides:
  - Secure Network Encryption Protocol (SNEP) for confidentiality, two-party data authentication, and data freshness
  - a "micro" version of the Timed, Efficient, Streaming, Loss-tolerant Authentication protocol (µTESLA) for authentication for data broadcast
  - assumption is that every node has a secret key shared with the base station

Security Protocols for Sensor Networks

- Secure Network Encryption Protocol (SNEP)
  - symmetric security (same message is encrypted differently each time)
  - replay protection
  - low communication overhead
  - uses MAC for two-party authentication and integrity
  - nodes A and B share a secret master key
  - master key used to derive four independent keys using pseudorandom function
    - two keys used for encryption of messages in each direction (K_{AB} and K_{BA})
    - two keys are used as message integrity codes (K'_{AB} and K'_{BA})

Security Protocols for Sensor Networks

- Secure Network Encryption Protocol (SNEP) (contd.)
  - complete encrypted message:
    A ➔ B: $\{D\}_{\{KAB,CA\}}$,MAC($K'_{AB}C_A || \{D\}_{\{KAB,CA\}}$)
  - D = data, K = key, C = counter, MAC computed as MAC($K', C || E$)
  - provides authentication (using MAC)
  - provides replay protection (using counter value in MAC)
  - freshness (counter value enforces message ordering); considered weak since sending ordering is enforced within node B, but no absolute assurance to node A that message was created by B in response to an event in A (nonce can be added to obtain strong freshness)
  - semantic security (counter is encrypted with each message, i.e., same message will be encrypted differently)
  - low communication overhead (counter state is kept at each end point and is not sent in message)

Security Protocols for Sensor Networks

- µTESLA
  - extension of TESLA protocol (by considering resource limitations)
  - focuses on need for authenticated broadcast in WSNs
  - relies on symmetric mechanisms provided by SNEP to authenticate first packet in broadcast message
  - TESLA uses digital signatures to authenticate initial packet and has an overhead of 24 bytes per packet
  - µTESLA emulates asymmetric cryptographic mechanism through a delayed disclosure of symmetric keys
  - µTESLA assumes that base station (BS) and sensor nodes are loosely time synchronized and each sensor knows upper bound on maximum synchronization error
  - when BS sends a message, it authenticates it by computing a MAC on the packet with secret key
  - when a node receives the packet, node knows that MAC key is only known to BS
  - node stores packet until the BS broadcasts the verification key to all receivers

TinySec

- Lightweight and generic link-layer security package
- Can easily be integrated into sensor network applications
- Supports two different security options:
  - authenticated encryption (TinySec-AE)
    - data payload is encrypted
    - MAC is used to authenticate packet
  - authentication only (TinySec-Auth)
    - entire packet is authenticated with MAC
    - payload is left unencrypted
- Relies on cipher block chaining (CBC) with specially formatted 8-byte initialization vector (IV) for encryption
- Relies on efficient and fast cipher block chaining construction (CBC-MAC) for computing and verifying MACs
  - using block cipher, number of cryptographic primitives that must be implemented is minimized
  - length of MAC is 4 bytes (attacker must try at most 2^{32} blind forgeries)

Localized Encryption and Authentication Protocol

- LEAP is a key management protocol for sensor networks, designed to support in-network processing
- Key observation is that different types of packets (control versus data) have different security requirements
- LEAP provides four keying mechanisms:
  - individual keys
    - every node has unique key shared with BS
    - key used for confidentiality and MAC
  - group keys
    - globally shared key used by BS to communicate with entire network
  - cluster keys
    - shared key between sensor and its neighbors
    - used for securing local broadcast messages
  - pairwise shared keys
    - shared key between sensor and one of its immediate neighbors

Localized Encryption and Authentication Protocol

- LEAP also provides a technique for local broadcast authentication
  - every node generates a one-way key chain of certain length
  - every node transmits the first key in the chain to each neighbor (encrypted with the pairwise shared key)
  - whenever a node sends a message, it takes the next key from the chain (each key is called an AUTH key) and attaches it to message
  - keys are disclosed in reverse order of their generation and a receiver can verify the message based on the first received key or a recently disclosed AUTH key

IEEE 802.15.4

- Four basic security models:
  - access control
  - message integrity
  - message confidentiality
  - replay protection
- Security is handled by the MAC layer
- Application can choose specific security requirements by setting appropriate parameters in the radio stack (default: no security)

IEEE 802.15.4: Security Suites

Name	Description
Null	No security
AES-CTR	Encryption only, CTR mode
AES-CBC-MAC-128	128-bit MAC
AES-CBC-MAC-64	64-bit MAC
AES-CBC-MAC-32	32-bit MAC
AES-CCM-128	Encryption and 128-bit MAC
AES-CCM-64	Encryption and 64-bit MAC
AES-CCM-32	Encryption and 32-bit MAC

ZigBee Security

- Introduces the concept of trust center (responsibility assumed by the ZigBee coordinator)
  - responsible for authentication of devices wishing to join network (trust manager)
  - responsible for maintaining and distributing keys (network manager)
  - responsible for enabling end-to-end security (configuration manager)
- Residential mode
  - trust center allows nodes to join network, but does not establish keys with the network devices
- Commercial mode
  - trust center generates and maintains keys and freshness counters with every device in the network
  - large memory cost

ZigBee Security

- ZigBee uses the CCM* mode for security, which is a combination of CTR mode and CBC-MAC mode
- Compared to CCM, CCM* offers encryption-only and integrity-only capabilities
- ZigBee has several levels of security, including:
  - no security
  - encryption only
  - authentication only
  - encryption and authentication
- ZigBee's MAC can vary from 4 to 16 bytes