IoT Security

Pervasive Computing and Wireless Sensor Networks

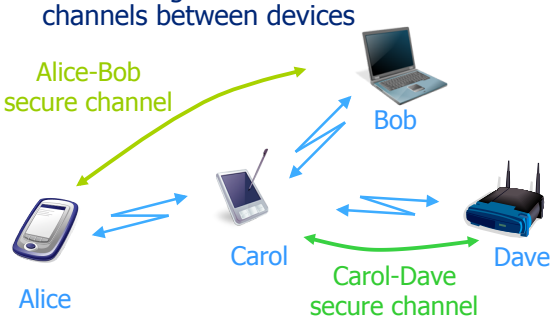
Panos Papadimitratos

Networked System Security Group

www.ee.kth.se/nss

Problem statement

- Establishing secure communication channels between devices



2

Problem statement (cont'd)

- Security requirements
  - Authentication
  - Integrity
  - Confidentiality
  - Non-repudiation
  - …

3

Problem statement (cont'd)

- Security mechanisms
  - Message Authentication Codes (MACs)
  - Digital signatures
  - Encryption/decryption
  - Passwords
  - …
- Cryptography
  - Asymmetric key
  - Symmetric key

4

Problem statement (cont'd)

- Enable secure communication
  - Uni-directional
  - Bi-directional
- Issues to consider
  - Long- or short- term?
  - What fraction of the system nodes?
  - Is there a trusted third party?
  - …

5

Public-key approach

Alice
Identity: A
Public key: K_A
Private key: k_A

Bob
Identity: B
Public key: K_B
Private key: k_B

- Advantage
  - Any-to-any secure communication
- Disadvantage
  - Need to bind public keys to identities

6

Public-key approach (cont'd)

Alice

(1) n_A, A, text, $Sig_A(n_A,A,text)$

Bob

(2) E_A(sec-text,B), $Sig_B(A,n_A,E_A$(sec-text,B))

- Secure communication example
  - Message (1): signed with k_A; n_A is a nonce
  - Message (2): sec-text and B encrypted with K_A; A, n_A, and ciphertext signed with k_B
  - Note: In practice, different keys are used for signing/ verifying and encrypting/decrypting

7

Public-key approach (cont'd)

Alice
Identity: A
Public key: K_A
Private key: k_A
$Cert_{CA}\{A,K_A\}$

Bob
Identity: B
Public key: K_B
Private key: k_B
$Cert_{CA}\{B,K_B\}$

- Certification Authority(CA)
  - Trusted Third Party
  - Known K_{CA}
  - $Cert_{CA}$: CA signature on the identity, public key, and other information (e.g., lifetime of the certificate)

8

Using a Certification Authority (CA)

- Largely independent of communication
  - Users can obtain certificates over the wire-line network
  - Certificates are installed at wireless devices and the corresponding keys are used to secure wireless communication
  - CA public keys should also be preinstalled
    - Similar to web browsers
- Examples specific to wireless networks

9

Using a CA (cont'd)

- Wireless local-area (e.g., campus-wide) networks
  - CA locally administered
  - IEEE 802.11 devices communicate securely with access points
- Tactical networks
  - CA operated by the corresponding government department
  - Keys and certificates installed at wireless-enabled devices
  - Hierarchical network organization
- Vehicular Communication (VC) Systems
  - Details in ANSS

10

Using a CA (cont'd)

- What does a certificate provide/contain?
  - Binding of public key and identity
  - Attributes of the subject (owner of the public key)
  - Certificate serial number
  - Information on the certificate issuer
  - Time of issuance
  - Lifetime (interval of validity)
  - Information on the used cryptography
  - Signature by the CA
- Internet standard: X.509

11

Using a CA (cont'd)

- Certificate revocation
  - Certificates can cease to be valid
    - Device or user is evicted
    - Corresponding cryptographic key is compromised
  - The CA publicizes which certificates are revoked
  - Certificate Revocation Lists (CRLs) are the most popular approach
  - Challenge: Update and distribute CRLs fast enough and to all interested parties

12

Public key cryptography - Practical aspects

- There is no single trusted authority
  - Nodes belonging to different administrative domains will in general be associated and execute security protocols
- Public key cryptography is feasible even in low-end mobile platforms, but it is costly
  - Processing
  - Energy consumption
  - Delays
  - Transmission overhead

13

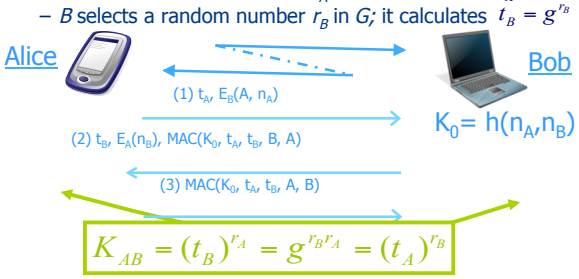
Symmetric key establishment

- Public key cryptography
  - Moderated use recommended
  - Examples:
    - Session keys
    - Shared symmetric key establishment
- Key agreement
  - Both nodes contribute to the shared symmetric key
- Key transport
  - One of the nodes 'chooses' the shared symmetric key

14

Key agreement

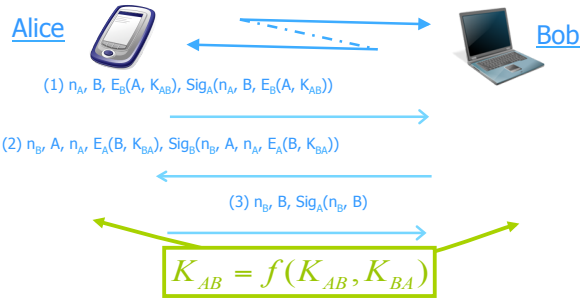
- Authenticated Diffie-Hellman protocol
  - *g* publicly known parameter; *G a* multiplicative group
  - *A* selects a random number r_A in *G;* it calculates $t_A = g^{r_A}$
  - *B* selects a random number r_B in *G;* it calculates $t_B = g^{r_B}$



H. Krawczyk, "SKEME: A versatile secure key exchange mechanism for Internet," NDSS'96

15

Key transport



X.509 three-pass key transport protocol

16

Key transport and establishment

- Can also be based on symmetric key cryptography
  - Key Distribution Centers (KDCs)
  - Kerberos
  - ISO/IEC 11770-2 standardized mechanisms
- How about reducing the cost?

17

Hash chains

- Cryptographic *hash* or *one-way* function
  - h : $\{0,1\}^* \rightarrow \{0,1\}^n$
  - Input: Arbitrary length
  - Output: Fixed length *n*
- Required properties
  - *Collision resistance*: it is computationally infeasible to find two distinct inputs, *x*, *y*, which hash to a common value $h(x)=h(y)$
  - *Pre-image resistance*: given a specific hash-value *z*, it is computationally infeasible to find an input *x* such that $h(x)=z$
  - *2nd pre-image resistance*: given *x* and $h(x)$ it is computationally infeasible to find a second input $y \neq x$ such that $h(y)=h(x)$
  - *Low computational cost*: given *h* and an input *x*, $h(x)$ is easy to compute.

18

Hash Chains (cont'd)

- Pick a random number *r*
- Generate *k* elements by hashing *r* successively *k* times

$$h^k(r) \leftarrow h^{k-1}(r) \leftarrow \mathrm{L} \leftarrow h^3(r) \leftarrow h^2(r) = h(h(r)) \leftarrow h(r)$$

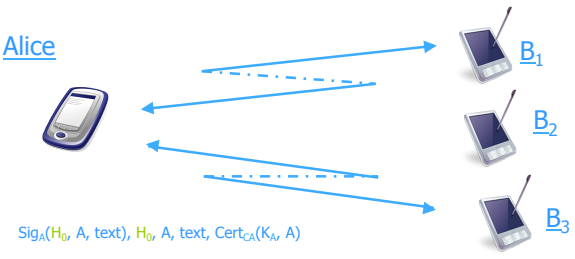
$$\| \quad \| \quad \| \quad \| \quad \|$$

$$H_0 \leftarrow H_1 \leftarrow \cdots \leftarrow H_{k-3} \leftarrow H_{k-2} \leftarrow H_{k-1}$$

- H_0 is the hash chain *anchor*
- The remaining *k-1* elements can be used for authentication

19

Bootstrapping a hash chain



- Alice must 'commit' to the hash chain anchor
- Each B_i node validates the commitment (signature) and stores H_0
- Alice can then utilize the hash chain elements

20

Using a hash chain

- Chain elements as authenticators, e.g., to transmit "yes" / "no"
  - "Yes" chain

$$H_0 \leftarrow H_1 \leftarrow \cdots \leftarrow H_{k-3} \leftarrow H_{k-2} \leftarrow H_{k-1}$$

  - "No" chain

$$G_0 \leftarrow G_1 \leftarrow \cdots \leftarrow G_{k-3} \leftarrow G_{k-2} \leftarrow G_{k-1}$$

Sender : 'Reveal' elements in this order

Use G_i or H_i to authenticate a "no" or "yes"

Receiver: For the i – th message from Alice, verify that $h^i(H_i) = H_0$ or $h^i(G_i) = G_0$

R. Hauser, A. Przygienda, and G. Tsudik, "Reducing the Cost of Security in Link-State Routing," NDSS'96

21

Using a hash chain (cont'd)

- Chain elements as symmetric keys

$$H_0 \leftarrow H_1 \leftarrow \cdots \leftarrow H_{k-3} \leftarrow H_{k-2} \leftarrow H_{k-1}$$

Time T_i : m_i = A, $text_i$, MAC(H_i, A, $text_i$)
Time T_{i+j} : Release H_i

  - Synchronized clocks at sender and receiver
  - Sender release keys (e.g., flooding them across the network) at specific intervals
  - *A posteriori* validation at the receiver: reject messages not generated sufficiently close to the release time

S. Cheung, "An Efficient Message Authentication Scheme for Link State Routing," Comp. Sec. App. Conf. '97

A. Perrig et al., "Efficient and secure source authentication for multicast," NDSS '01

22

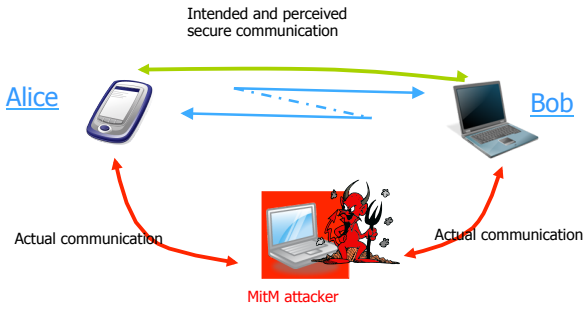
Recap: Public key enabled security

- Advantages
  - Any-to-any secure communication
  - Basis for bootstrapping symmetric key primitives
- Disadvantages
  - Processing and communication overhead
  - Setting up a certification authority
- Comment
  - Methods discussed so far are rather 'agnostic' to the underlying network technology

23

What if no CA is available?

- **Main challenge:** *Man-in-the-Middle* attacks



24

What if no CA is available? (cont'd)

- Can we leverage on characteristics of the network or the mobile application?
- **Observation 1:** Wireless, mobile devices are used by human beings, who can assist the security association establishment
- **Observation 2:** Wireless communication possible only within a very short range or within a line of sight can imply that no other device is present (**caution!**)

25

Leveraging on the users

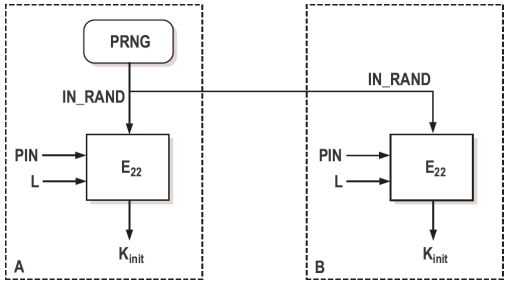
- Bluetooth
  - Short-range communication
  - Master-slave configuration
  - Frequency hopping
  - Security issues
    - Mutual device authentication
    - Confidentiality

Reading reference: Security Overview of Bluetooth, by Dave Singelee, Bart Preneel, URL: http://www.cosic.esat.kuleuven.be/publications/article-565.pdf

26

Leveraging on the users (cont'd)

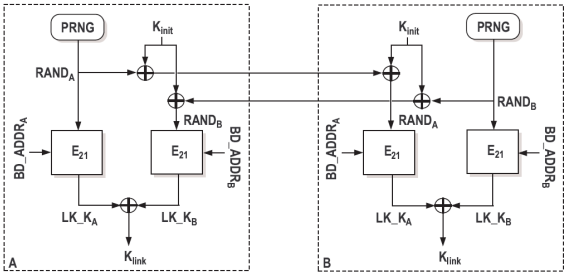
- Bluetooth initialization key establishment



27

Leveraging on the users (cont'd)

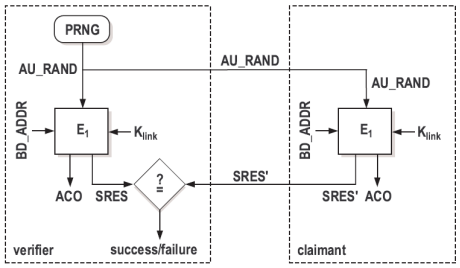
- Bluetooth link key establishment



28

Leveraging on the users (cont'd)

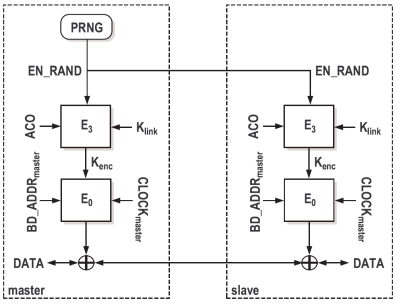
- Bluetooth authentication



29

Leveraging on the users (cont'd)

- Bluetooth encryption key generation



30

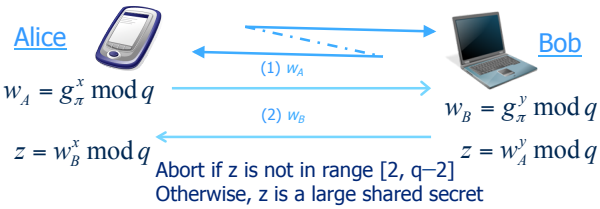
Leveraging on the users (cont'd)

- Weaknesses
  - The PIN length determines how difficult it is break the protocol
    - Four (4) digit numbers are easy to guess
    - Brute force attack requires to try out $10^4 = 10000$ times
    - Off-line cracking
  - Worse even: often, only the default PIN is used
  - For memory-constrained devices: the link key is the long-term unit key of the device
  - The E_0 stream cipher has weaknesses

31

Leveraging on the users (cont'd)

- Password-based key establishment
  - π : shared password
  - $g_\pi = (h(\pi))^2 \bmod q$
  - q publicly known parameter
  - *A, B* select random numbers *x, y respectively*

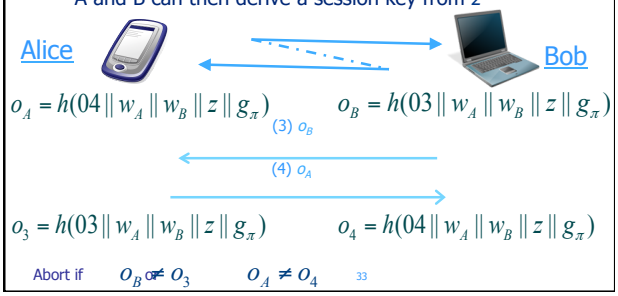


D. Jablon, "Extended Password Key Exchange Protocols Immune to Dictionary Attacks," WET-ICE '97

32

Leveraging on the users (cont'd)

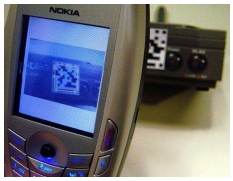
- Password-based key establishment
  - h : hash function
  - Once *z* is established, A and B prove to each other they know the same *z*
  - A and B can then derive a session key from *z*

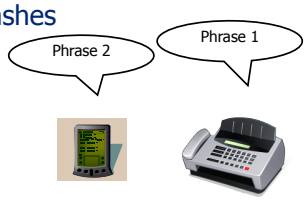


33

Leveraging on the user (cont'd)

- The user verifies that the keys generated at the two devices are identical
- Visual and audible hashes





J. McCune, A. Perrig, and M. Reiter, "Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentiction." S&P'05

M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud And Clear Human-Verifiable Authentication Based on Audio", ICDCS'06

34

Leveraging on the wireless link

- 'Off-line' local channels
  - One example: infra-red
    - D. Balfanz, D. Smetters, P. Stewart, and H. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," NDSS'02
  - Exchange information over the local channel that allows you to authenticate over the wireless radio channel
- Caution: System and protocol design must ensure that it is indeed impossible for the attacker to interfere actively with the communication over the local channel
  - For example, the attacker must be unable to act as an 'invisible' relay

35

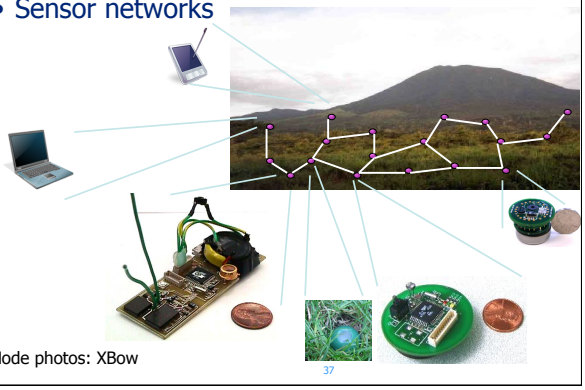
Leveraging on the network

- Mobility
  - Users meeting each other, e.g., at a conference, can set up symmetric keys or exchange public keys
    - S. Capkun, J-P. Hubaux, and L. Buttyan, "Mobility helps security," ACM Mobihoc'03
  - More generally, a mobile device can be interested in obtaining public keys of other devices in proximity, e.g., within a few hops
    - Example later in secure routing
  - Point of caution: communication pattern

36

Wireless Sensor Networks

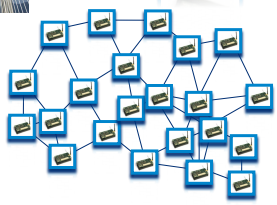
- Sensor networks



Node photos: XBow

37

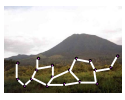
Wireless Sensor Networks (cont'd)



Not yet wireless…

38

Wireless Sensor Networks (cont'd)



Context awareness

Sensing

Graphics by Nokia

39

What can go wrong?

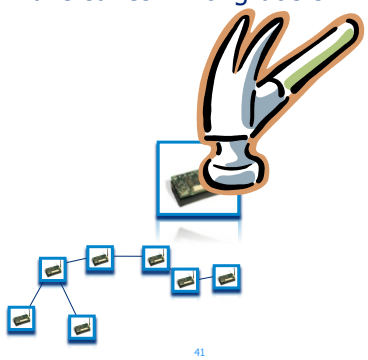
- Run out of battery
- Break down
- Radio interference
- Obstacles

- Sounds natural…

40

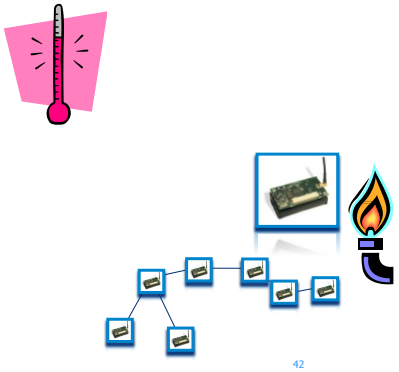
What can go wrong? (cont'd)

- Attackers! Adversaries! Wrong-doers!



41

What can go wrong? (cont'd)



42

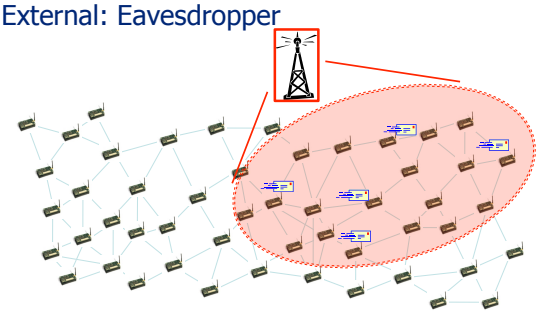
What can go wrong? (cont'd)

- Let's think
  - Bits
  - Bytes
  - Signals
  - Antennas
  - Software
- The adversary
  - Comes in with own devices
  - Takes over network devices
  - Or both!

43

Attacking WSN

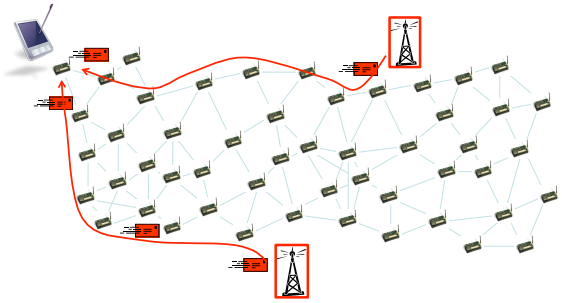
- External: Eavesdropper



44

Attacking WSN (cont'd)

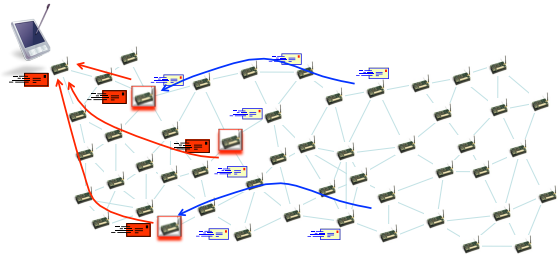
- External: Message/measurement injection



45

Attacking WSN (cont'd)

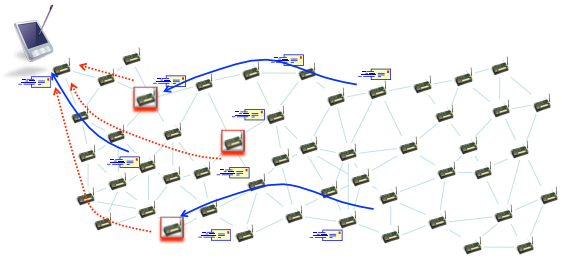
- Internal: Packet modification



46

Attacking WSN (cont'd)

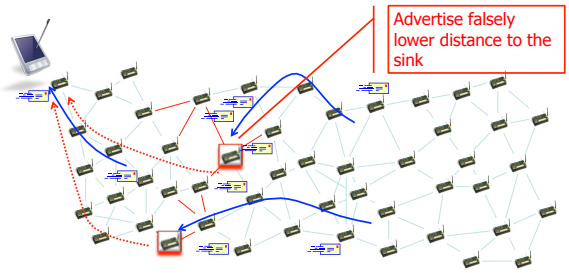
- Internal: Packet dropping



47

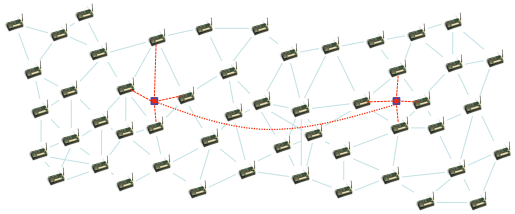
Attacking WSN (cont'd)

- Internal: Packet dropping (cont'd)



48

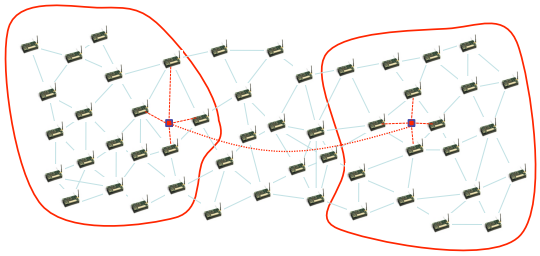
Attacking WSN (cont'd)



- External: Attacking neighbor discovery
  - Create fictitious links
  - *Wormhole attack* : adversaries 'transport' traffic

49

Attacking WSN (cont'd)

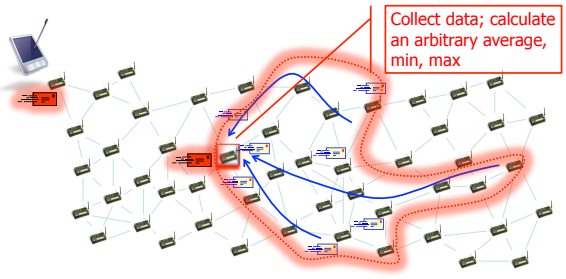


- External: Attacking neighbor discovery
  - Controlling eventually communication

50

Attacking WSN (cont'd)

- Internal: Control/corrupt aggregation



51

Attacking WSN (cont'd)

- Combinations of the above?
  - Yes, e.g., internal nodes that intercept data
  - …
- Variations?
  - Yes, e.g., stealthy attacks, or hitting when it hurts the most'
  - Mobile, adaptive attacks
  - …
- Other attacks?
  - Sure! E.g., detecting the area of a sensed event
  - Or jamming
  - …

52

Attacking WSN (cont'd)

- How do the attacks take place?
- The adversary
  - Brings in own devices
  - Installs own devices/nodes in the system
  - Compromises devices/nodes already in the system
    - Getting physical access
    - Injecting malware

53

Attacking WSN (cont'd)

- What does this mean?
- The legitimate user may:
  - Loose control of the network operation
  - Loose part or all of the data
  - Get meaningless/fake data
  - Have data stolen
  - Have applications based on the WSN manipulated

We have to prevent all that!

54

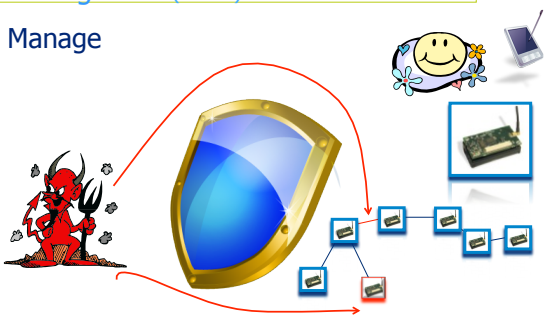
Securing WSN

- Prevent

55

Securing WSN (cont'd)

- Manage



56

Key Management

- Public key methods are rather expensive
  - Even though several implementations are available now
- Symmetric key approaches predominant
  - Caution: Symmetric keys grow fast in numbers; for a network of N nodes, N^2 symmetric keys
  - Maybe less, depending on the utilized secure(d) protocols

57

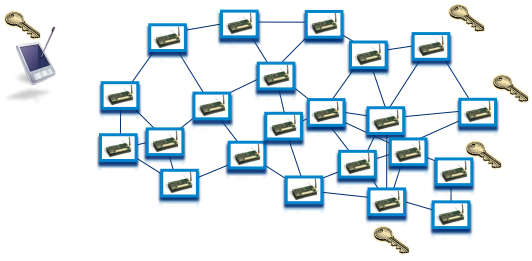
Key Management (cont'd)

- Questions: How many keys, shared by which nodes, why, how, for how long?
- Answers:
  - Key Distribution (or Pre-Distribution)
  - Key Discovery
  - Key Establishment
  - Key Update

58

Key Management (cont'd)

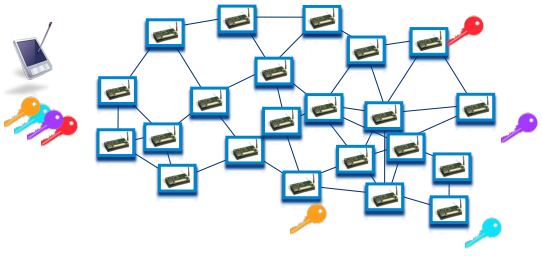
- One key for all
  - Low security



59

Key Management (cont'd)

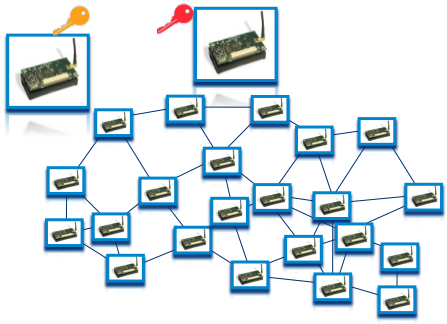
- One key per node
  - Shared with the sink



60

Key Management (cont'd)

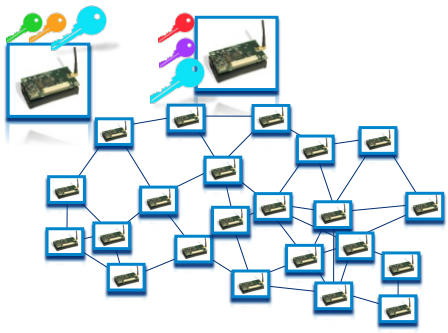
- What about node-to-node authentication?



61

Key Management (cont'd)

- What about node-to-node authentication?



62

Key Management (cont'd)

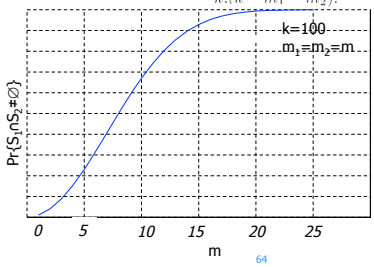
- Initialization
  - Large pool S of unique keys are picked at random
  - Each node is preloaded with m keys selected randomly from S
- Key discovery phase
  - Which neighbors do I share a key with?
- Key establishment (if needed)
  - Establish a key with a neighbor through another neighbor
- How many keys should each node get?

63

Underlying idea

- Dimension (number of keys) the *key ring* of each node (m_1, m_2) and the *key pool* (k), so that probability that any two nodes share at least one key gets the sought value

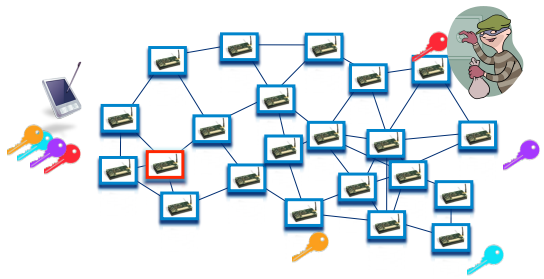
$$\Pr\{S_1 \cap S_2 \neq \emptyset\} = 1 - \frac{(k-m_1)!(k-m_2)!}{k!(k-m_1-m_2)!}$$



64

WSN Security – One example

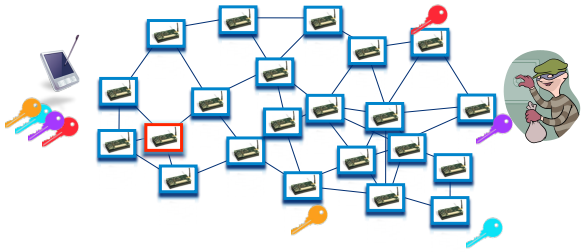
- Parasitic adversary



65

WSN Security – One example

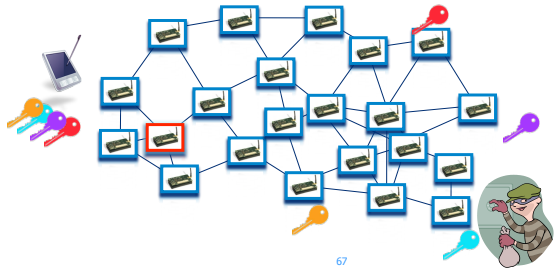
- Parasitic adversary



66

WSN Security – One example

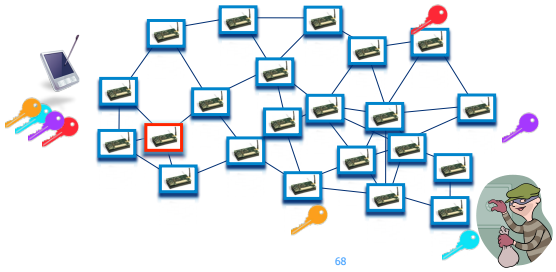
- Parasitic adversary



67

WSN Security – One example

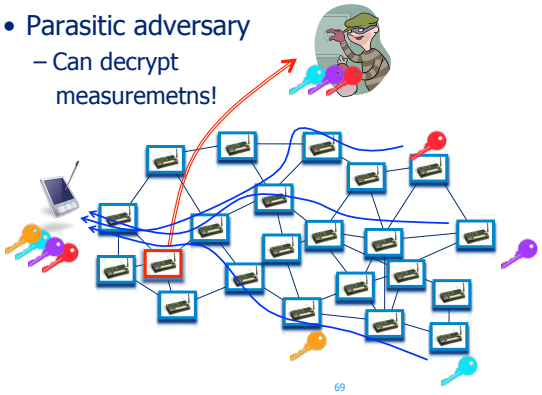
- Parasitic adversary



68

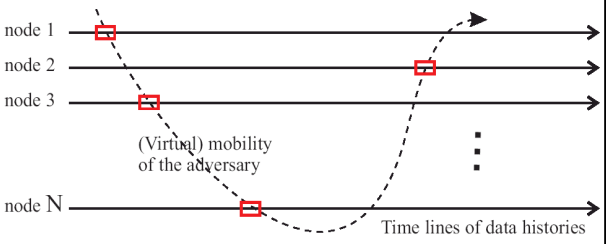
WSN Security – One example

- Parasitic adversary
  - Can decrypt measuremetns!



69

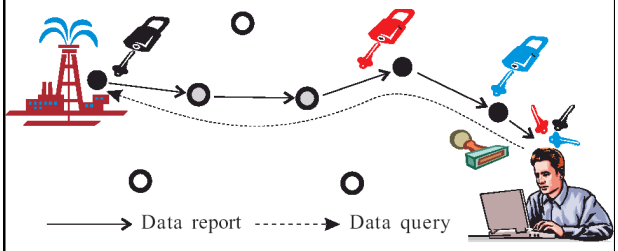
Parasitic adversary (cont'd)



- Powerful
- Realistic
- Note: cannot modify the functionality of nodes

70

En route re-encryption



- Source node always encrypts its data
- Intermediate nodes "flip a coin" and re-encrypt packets with probability *q*

71

Key refreshing

- Sensor nodes randomly elect a point in time to execute the key refreshing protocol
  - The time for the protocol execution is drawn randomly around a system specific parameter for the average refresh rate
  - The sensor node generates a new symmetric key
  - It encrypts it with the sink public key
  - It transmits it as if it were a regular data measurement

72

Key refreshing (cont'd)

- The sink does *not* know which node is physically compromised
  - It can only guess which part of the network the adversary may target
  - It is still clueless among that subset of nodes which are compromised
- The adversary does not know which node is refreshed
  - It could guess if symmetric key transport was used, and
    - It were within one hop from the refreshed node
    - It had compromised the entire path from from the source
- The adversary can always re-read the key of previously compromised nodes

73

Questions to be answered

- What is the probability that a piece of data is always is encrypted with at least one symmetric key the adversary does not have?
- What is the probability that the adversary breaches confidentiality?

P. Papadimitratos, J. Luo, and J.-P. Hubaux, "Randomized Countermeasure Against Parasitic Adversaries in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications* (*IEEE JSAC)*, Vol. 28, No. 7, pp. 1036 – 1045, September 2010

74

Other important questions

- How to implement security?
  - Esp. given the constraints of sensor nodes
- How to secure other types of WSN functionality?
  - In other words, how to stop the attacks we discussed earlier?
- How to prove security?
- How to secure WSN when with other systems? E.g., vehicular communication systems?

75