

POSH: Proactive co-Operative Self-Healing in Unattended Wireless Sensor Networks

Roberto Di Pietro, Di Ma, **Claudio Soriente** and Gene Tsudik

UNESCO Chair in Data Privacy
Universitat Rovira i Virgili

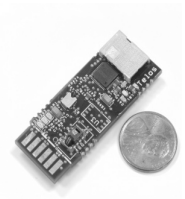
Computer Science Department
University of California, Irvine

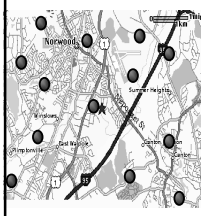
SRDS 2008, Napoli

Agenda

- Introduction to UWSNs
- Motivation
- POSH
- Related Work
- Summary

“Typical” Wireless Sensor Network





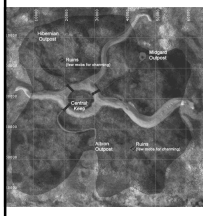


Many real, alleged and imagined applications

- Networking
  - Sensor-to-sink communication (opt. sink-to-sensors)
- Collection method
  - Periodic collection
    or
  - Event driven
    or
  - Query based = on-demand
- Online Sink
  - Real-time off-loading of data

Unattended Wireless Sensor Network (UWSN)





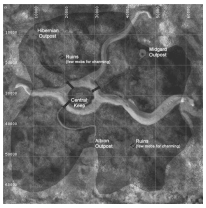


- UWSNs operate in hostile environments
  - Initial deployment might be ad-hoc
- No ever-present sink
  - Itinerant, visits UWSN periodically
- Periodic data sensing
  - Nodes might retain data for a long time
  - Data might be valuable
- Nodes are mostly left on their own
  - Adversary roams around with impunity
  - Adversary has **lots of time**
- **Challenge: Data Security**
  - **Confidentiality**
  - **Authenticity/Integrity**
  - **Survival/Availability**

Examples



- WSN in a recalcitrant country monitoring nuclear activity



- Underground WSN monitoring sound and vibration produced by troop movements or border crossings



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

New kind of Adversary (ADV)

- Previous adversaries corrupt fixed number of nodes over all time (k out of n)
  - Solutions focused on detection
  - Once attack is detected, on-line sink can help
    - e.g., exclude compromised nodes
- Our adversary is MOBILE
  - Roams the network, compromises k-out-of-n nodes per round
    - Given enough time, can subvert the entire UWSN
  - Sink is off-line: real-time detection does not help
    - ADV can come and go with impunity

Motivation

- Curious adversary wants to read/learn sensor-collected data
- Encryption does not help
  - Symmetric keys are exposed with node compromise
  - With public key encryption, ADV can GUESS the plaintext
    - Randomized public key encryption helps but only with a TRNG
    - Not currently available (nor foreseeable)
- Can we protect category (1) and (3) data?

Sensor data can be partitioned, based on time of compromise:
(1) Collected Before Compromise
Need Forward Secure Encryption
(2) Collected During Compromise
Nothing helps…
(3) Collected After compromise
Need Backward Secure Encryption

Forward Secrecy

- Even if ADV learns the current key, it cannot derive PREVIOUS round keys
- Per-round key evolution
  - At the end of round r, next round key computed using one-way function (current key is securely erased)

$$K^{r+1}=H(K^r)$$

- But, after compromise, ADV can mimic key evolution process



$K^1 \rightarrow K^2 \rightarrow K^3 \rightarrow K^4 \rightarrow K^5 \rightarrow K^6 \rightarrow K^7 \rightarrow \ldots$

Sensor compromised at round 4 and then released



$\ominus \leftarrow K^4 \rightarrow K^5 \rightarrow K^6 \rightarrow K^7 \rightarrow \ldots$

Backward Secrecy

- Even if ADV learns the current key, it cannot derive FUTURE round keys
- Also, based on per-round key evolution
  - Requires online TTP, or secure hw at each node (same as distributed TTP)
- Not suitable for UWSNs
  - Our sink is off-line
  - A single sensor cannot act as a TTP for peers since it can be compromised at any time



$K^1 \rightarrow K^2 \rightarrow K^3 \rightarrow K^4 \rightarrow K^5 \rightarrow K^6 \rightarrow K^7 \rightarrow \ldots$

Sensor compromised at round 4 and then released



$K^1 \leftarrow K^2 \leftarrow K^3 \leftarrow K^4 \rightarrow$ ⛔

Key Insulated Secrecy

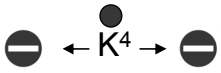
- Schemes that are both BACKWARD and FORWARD secure are known as KEY INSULATED schemes
  - No such scheme available for UWSNs
  - Require online trusted third party
  - Expensive computation



$K^1 \rightarrow K^2 \rightarrow K^3 \rightarrow K^4 \rightarrow K^5 \rightarrow K^6 \rightarrow K^7 \rightarrow \ldots$

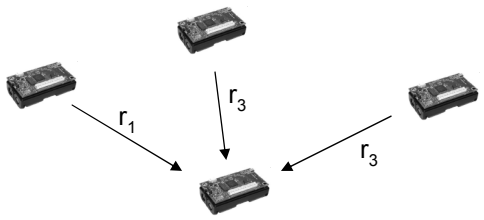
Sensor compromised at round 4 and then released





POSH: Main Idea

- Forward secrecy through key evolution
- Backward secrecy through sensor cooperation
  - A sensor can securely generate a key unknown to ADV, if it obtains at least one contribution from a non-compromised peer



$K_{new} = F(K_{old}, r_1, r_2, r_3)$

Network Assumptions 1/2

- Unattended scheduled (per round) data sensing/collection
  - Max ***v*** rounds between sink visits
  - Each sensor collects one data item per round
  - ADV's round = UWSN round
- Communication
  - UWSN always connected
  - Any two sensors can communicate either directly or through peers
- Storage
  - Each sensor has at least O(v) storage

Network Assumptions 2/2

- Cryptographic Capabilities
  - Hashing, e.g., SHA-2
  - Symmetric encryption
    - Unique secret key shared with the sink
  - Pseudo-Random Number Generator (PRNG)
    - Unique secret seed shared with the sink
- Re-initialization
  - At each visit, sink re-initializes ALL sensors (ADV not present):
    - New secret key
    - New secret seed
    - Empty storage

Adversarial Model

- Goal
  - Learn as many secret as possible (Why?)
- ADV compromises at most ***k*** (out of ***n***) nodes per round
  - Compromised nodes not necessarily contiguous/clustered
  - Reads state/storage
  - Listens to all incoming/outgoing communication
- Minimal disruption
  - ADV does not interfere with sensor behavior (e.g., in order to stay undetected)

POSH: Algorithm

Generic node i protocol at round r:

1. Generate t random values $\{R_{i_1}, \ldots, R_{i_t}\}$
2. Select $\{s_{i_1}, \ldots, s_{i_t}\} \leftarrow_R \{s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n\}$
3. Send R_{i_j} to s_{i_j}, $1 \leq j \leq t$
4. Receive contributions $\{c_{i_1}, \ldots, c_{i_{t'}}\}$
5. Sensing, encryption, authentication...
6. Compute $K_i^{r+1} = H(K_i^r || c_{i_1} || \ldots || c_{i_{t'}})$
7. Erase K_i^r

Contributions

Nodes to contribute to

Normal operation activities

Key refresh

$\{s_1, \ldots, s_n\}$ = set of sensors in the network
K_i^r = key used by s_i at round r

Analysis (aka Sensor Coloring)



Red sensors (R^r)

- currently controlled by ADV
- K per round



Yellow sensors (Y^r)

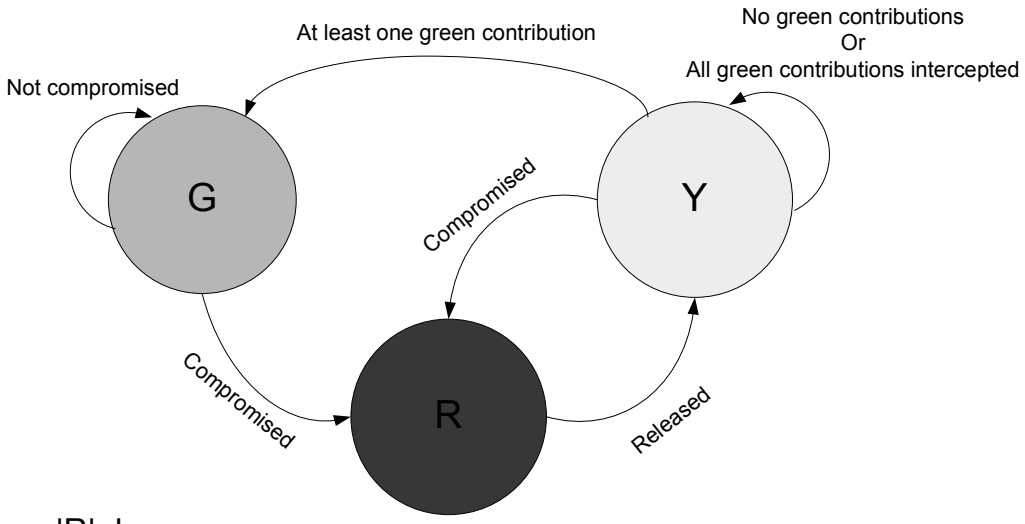
- Compromised in some previous round
- Their current keys are known to ADV
- At least k per round
  - released during previous round



Green sensors (G^r)

- Either never been compromised
- **Or** recovered through POSH

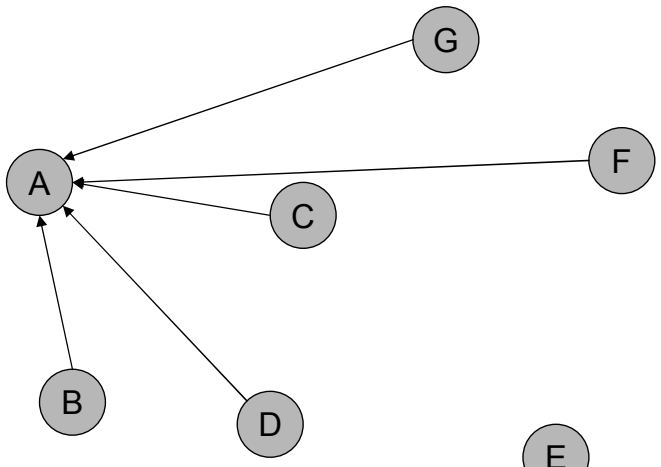
Sensor transition diagram



- |R|=k
- ADV's goal it to maximize |Y|+|R|
- Network goal: |G|=n-2k

Example

r = 3



Sensor A

K^1

$K^2=H(K^1 \| c_C \| c_F)$

$K^3=H(K^2 \| c_B)$

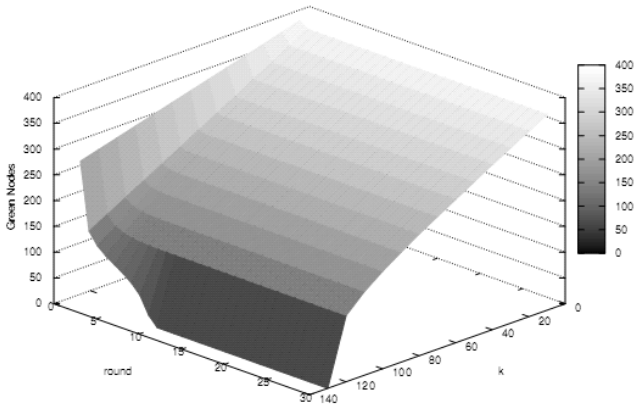
$K^4=H(K^2 \| c_D \| c_G)$

Two kinds of ADV

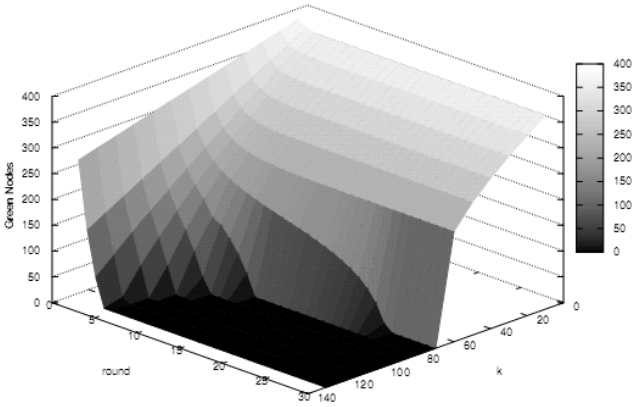
- INF-ADV is always aware of G
  - Unrealistic but very powerful
  - Never compromises a sensor if it of no use
  - Used as benchmark

- RR-ADV compromises sets of k nodes in a round-robin fashion
  - Time-based heuristic
    - Nodes that have been in Y for a long time could have since moved to G
  - Realistic, but possibly weak
  - Might compromise yellow sensors

Results – |G| vs. INF-ADV



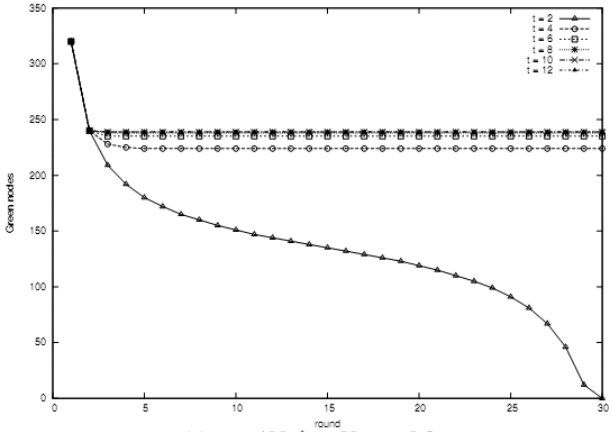
(a) $n = 400, t = 6, p = 0.2$



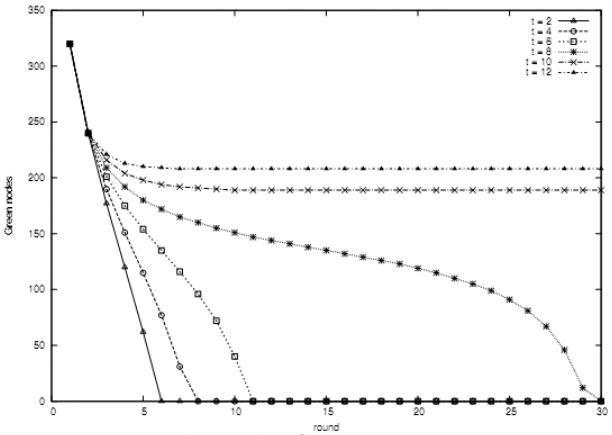
(b) $n = 400, t = 6, p = 0.8$

- p = ADV eavesdropping prob.
- t = 6 results in each sensor receiving at least one contribution, on average
- Threshold phenomena:
  - e.g. for p=0.2, |G| remains stable for k/n < 80/400
  - That is 20% per round!!!

Effect of “t”



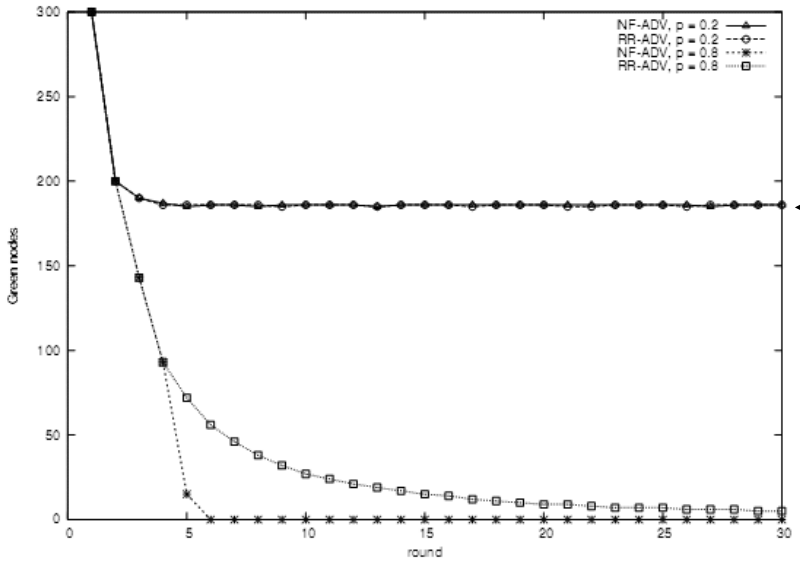
(a) $n = 400, k = 80, p = 0.2$



(b) $n = 400, k = 80, p = 0.8$

- Increasing t when |G| ~ n-2k does not help
- Also, messages are expensive!

INF-ADV vs. RR-ADV



No difference if |G| is close to its optimal value

$$n = 400, k = 100, t = 6$$

Dealing with the “real world”

- Message delivery failure
  - Sink synchronization
  - Sensors must store the IDs of their contributors
- Sensor failure
  - If a sensor dies, key evolution history (of other sensors) cannot be reconstructed
  - Other sensors might depend on the failed node
- Public Key Crypto
  - Encrypt round key under the sink PK
    - Use round key for everything else
    - Why is this secure and robust?

Related Work

- WSN Key distribution
  - Random key pre distribution
    - [Eschenauer and Gligor, CCS 2002]
    - [Chan, Perrig and Song, S&P 2003]
    - [Du, Deng, Han, Varshney, Katz andKhalili, Trans. Inf. Sys. Secur. 2005]
    - [Di Pietro, Mancini, Mei, Panconesi, Radhakrishnan, Trans. Inf. Sys. Secur. 2005]
  - Multi-stage deployment
    - [Castelluccia et al., Securecomm 2007]
  - Backward + Forward Secrecy in WSN
    - [Naik, 2003, IEEE Distributed Systems Online 2003]
- UWNS
  - Model definition
    - [Di Pietro, Mancini, Spognardi, Soriente and Tsudik, PerCom 2008]
  - Key resilience through sensor cooperation
    - [Ma and Tsudik, SSS 2008]

Summary

- UWSN is a new, exciting field that calls for innovative security solutions
- Sensor cooperation is essential to achieve security against a powerful adversary
- Robustness issues
- Role of randomization in UWSN not completely characterized yet

Questions?

Thank you!