Security and Privacy
in Unattended Sensor Networks
(or How to Cope with a Mobile Adversary)

Gene Tsudik
SPROUT: Security & Privacy Research Outfit
UC Irvine
http://sprout.ics.uci.edu

Joint work with:

Roberto Di Pietro
Università di Roma Tre

Claudio Soriente
Polytechnic University of Madrid

Luigi Mancini
Università di Roma "La Sapienza"

Gabriele Oligeri
Università di Pisa

Angelo Spognardi
Università di Roma "La Sapienza"

Di Ma
University of Michigan, Dearborn

Current Research

http://sprout.ics.uci.edu

- Privacy-Preserving Techniques
  - PSI, PPIT and Secret Handshakes
  - Location Privacy in MANETs/VANETs
  - Private Querying in WSNs
- Security of Personal and Embedded Devices
  - Unattended WSNs
  - Security for Embedded Devices
  - Distance Bounding
  - Usable Security (wireless device pairing, personal RFIDs)
- Internet Security
  - Privacy-agile Name Service
  - Phishing and Typo-squatting Countermeasures
  - DTN security

Security Research in General

- Reactive
  1. Identify existing security problem and adversary
  2. Suggest fixes

  OR:
  1. Spot problems in existing solutions
  2. Expose them

- Proactive: a 4-step process…

Step 1: Invent *plausible* and *scary* new adversary

Step 2: If needed, postulate new exciting (and *viable*) habitat for scary new adversary

Step 3: Develop *credible, effective and practical* weapons against adversary

Step 4: Market your fairy tale

Roadmap

- Introduction & Motivation
- Naïve defense strategies
- Cryptography?
- Distributed Self-healing
- (if time permits) Mobility & Attestation
- Conclusions

Wireless Sensor Networks

Many real, alleged and imagined applications

- Networking
  - Sensor-to-sink communication (opt. sink-to-sensors)
- Collection method
  - Periodic collection

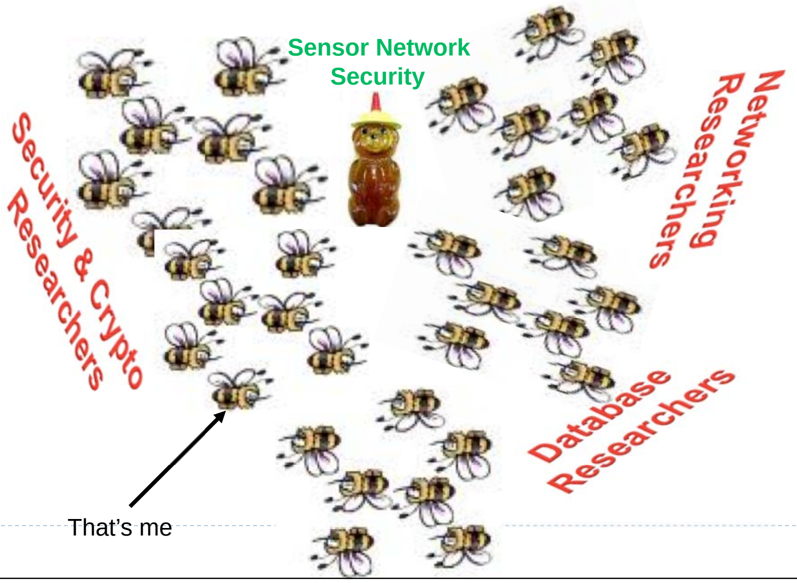
    or
  - Event driven

    or
  - Query based = on-demand
- Online Sink
  - Real-time off-loading of data

Lots of Prior Work

Sensor Network
Security

Lots of Prior Work



Sensor Network
Security

Security & Crypto
Researchers

Networking
Researchers

Database
Researchers

That's me

Recent WSN Security Topics

- Key management
- Secure routing
- Secure broadcasting/multicasting
- Secure querying
- Secure data aggregation / statistics
- Efficient cryptographic primitives
- Various attacks counter-measures, e.g. denial-of-message, cloning, sleep deprivation…

Prior Work on WSN Security

- Almost all prior work (pre-2008) assumed that the WSN is supervised by a TTP/Collector/Sink/Base-Station/etc.
- Is this always so?
- What if WSN is unattended most of the time?

Unattended
Wireless Sensor Network (UWSN)

- Hostile deployment environment
- No constantly present sink
  - Itinerant, visits periodically
- Periodic data sensing
  - Nodes might retain data for a long time
  - Data is valuable
- Nodes are mostly left on their own
  - Adversary roams around with impunity
  - Adversary has **lots of time**
- **Challenge: Data Security**

Examples

- WSN deployed in a recalcitrant country to monitor 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

UWSN Mobile Adversary (1)

Adv defined by: goal + operation + visibility

Goal:

- Targeted
  - Search-and-erase
  - Search-and-replace
- Non-targeted
  - Curious
  - Polluter
  - Eraser

Operation:

- Reactive
- Proactive

Visibility:

- Stealthy
- Visible

UWSN Mobile Adversary (2)

- Well-informed
  - Knows network topology and network defense strategy
- Mobile
  - Migrates between sets of nodes between sink visits
- Erratic
  - Unpredictable and possibly untraceable movement
- Data-centric
  - No interference with sensing or network operation
- Powerful but not omnipotent
  - Compromises up to a fixed # (k out of n) of nodes

UWSN Mobile Adversary (3)

- Previously considered adversaries capable of corrupting fixed number of nodes (k) overall
  - Solutions focused on detection
  - Once detected, on-line sink can mitigate attacks
    - e.g., exclude compromised nodes
- Our adversary is ***MOBILE***
  - – Roams and compromises different subsets of sensors
    - Given enough time, can subvert entire WSN
  - – Sink is off-line: real-time detection does not help
    - Adv can reach its goal and leave with impunity (remain undetected)

Assumptions

- Scheduled (per round) data sensing/collection
  - Max ***v*** rounds between sink visits
  - Adv round = UWSN round
- Adv compromises at most ***k*** (out of ***n***) nodes per round
  - Compromised nodes not necessarily contiguous
  - Reads all storage/memory
  - Listens to all incoming and outgoing communication
- Adv knows what data to target and when it was sensed
  - Receives external signal at collection time
    - Target node identity + collection round
    - Possibly knows the target value
- UWSN knows nothing… ➔ Equal protection for all data

This might sound familiar

Cryptographic Mobile Adversary in Proactive Threshold Cryptography [1]

- Proactive Cryptography: Decryption and Signatures (e.g., RSA, DSA)
- Adversary wants to learn a shared global secret
  - Corrupts at most k out of n nodes per round
  - Moves atomically at end of each round
- Our setting is different
  - No global secret
  - Meager resources
  - New solutions required

[1] Ostrovsky & Yung, How to Withstand Mobile Virus Attacks, PODC 1991
  - And lots of related literature since then…

END PART 1

Stealthy Search-and-Erase Adv

IEEE Percom'08

What if sensors have no crypto capability?

- Ultra-cheap sensors
  - No cryptographic abilities
  - Can only try to hide data location

- Data Migration strategies
  - Move Once
  - Keep Moving

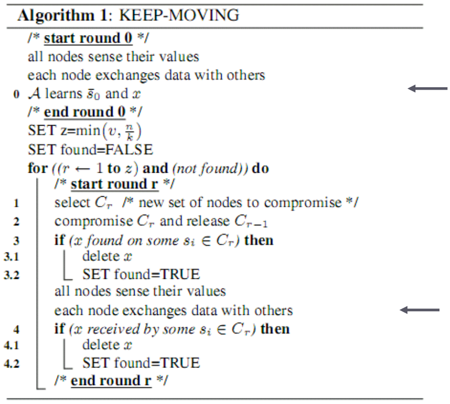
- Adv Goal: Search-and-erase
  - Looks for target data in compromised sensors

- Adv strategy:
  - Lazy
  - Frantic
  - Smart

Move Once

- Data off-loaded to a random peer recipient
  - Kept there for subsequent rounds (<v), until sink visit
- Adversary wins in at most $\left\lceil \frac{n}{k} \right\rceil$ rounds
  - Round 0
    - Learns originating node (data not there any longer)
  - Round i
    - Move to next set
  - At most $\left\lceil \frac{n}{k} \right\rceil$ rounds to find and erase

Keep Moving



```
Algorithm 1: KEEP-MOVING
    /* start round 0 */
    all nodes sense their values
    each node exchanges data with others
0   $\mathcal{A}$ learns $\bar{s}_0$ and $x$
    /* end round 0 */
    SET z=min$(v, \frac{n}{k})$
    SET found=FALSE
    for (($r \leftarrow 1$ to $z$) and (not found)) do
        /* start round r */
1       select $C_r$  /* new set of nodes to compromise */
2       compromise $C_r$ and release $C_{r-1}$
3       if ($x$ found on some $s_i \in C_r$) then
3.1         delete $x$
3.2         SET found=TRUE
        all nodes sense their values
        each node exchanges data with others
4       if ($x$ received by some $s_i \in C_r$) then
4.1         delete $x$
4.2         SET found=TRUE
        /* end round r */
```

Adv learns target data at round 0

Adv looks for target data in the new set of compromised nodes

Nodes exchange messages

Adv looks for target data in the messages received by corrupted nodes

- Adv has two chances per round
  - Before data exchange
  - After data exchange

Keep Moving – Lazy

- Exploit data being always on the move
- Two chances at round 1; one chance each new round
- Prob. data survives to v rounds

$$P_L(v) = P_1 \cdot P_2^{v-1}$$

$$P_1 = \frac{k}{n} + \left(1 - \frac{k}{n}\right)\frac{k}{n} = \left(1 - \frac{k}{n}\right)^2 \qquad P_2 = 1 - \frac{k}{n}$$

Keep Moving – Frantic

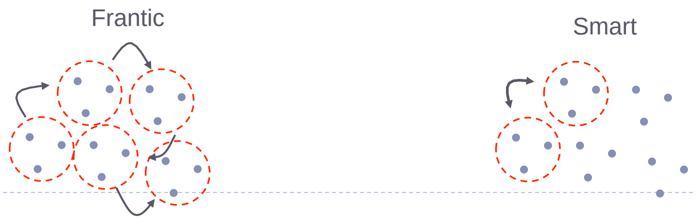
- Select a new random k-set at each round
- Two chances per round
- Probability that data survives v rounds:

$$P_F(v) = P_1 \cdot P_2^{v-1} \cdot P_3^{v-1}$$

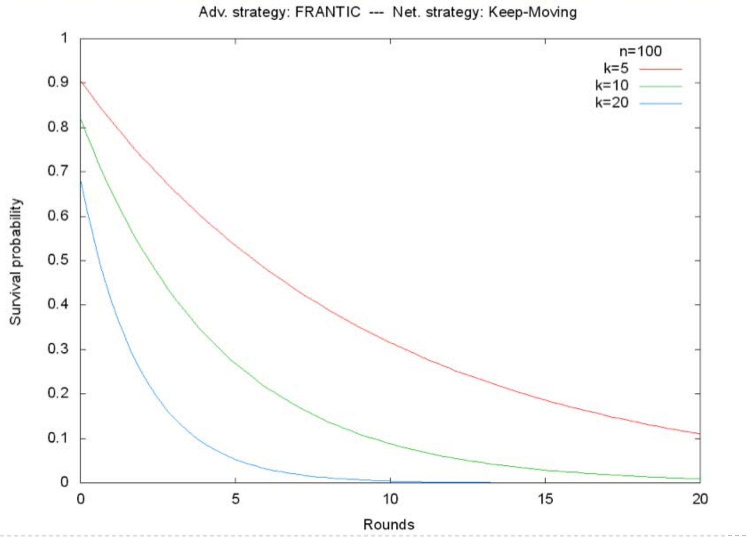
$$P_1 = \frac{k}{n} + \left(1 - \frac{k}{n}\right)\frac{k}{n} = \left(1 - \frac{k}{n}\right)^2 \qquad P_2 = 1 - \frac{k}{n} \qquad P_3 = 1 - \frac{k}{n-k}$$

Keep Moving – Smart

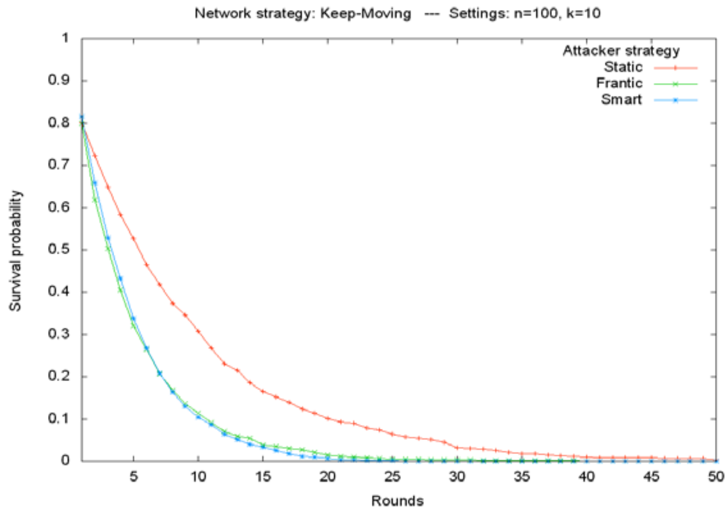
- Moves between two fixed (non-overlapping) set of nodes
  - No matter what adversarial strategy, data recipient node is always chosen according to an uniform distribution
  - Same survival probability!



Results


Adv. strategy: FRANTIC --- Net. strategy: Keep-Moving

Keep Moving



Replication

- Each sensor produces R copies of each data
  - Data survives as long as one copy survives
- $X_{i,j}$ = 1 if replica i survives up to round j

$$Pr[X_{1,j}=1]=P_1\cdot P_2^{j-1}\cdot P_3^{j-1}$$

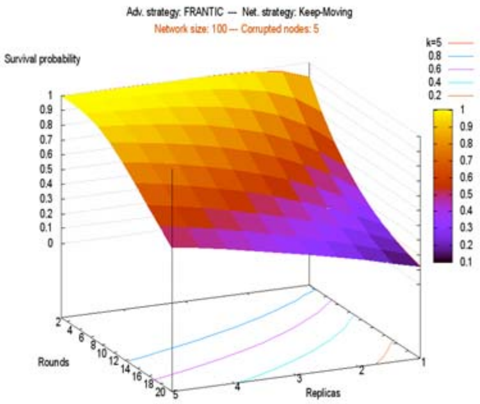
$$\overline{P_R^v}=Pr[X_{1,v}=0\wedge\ldots\wedge X_{R,v}=0]=Pr[X_{1,v}=0]^R=$$

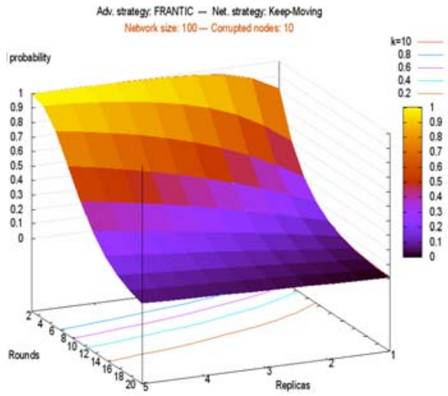
$$=(1-Pr[X_{1,v}=1])^R=(1-P_1\cdot P_2^{v-1}\cdot P_3^{v-1})^R$$

- Prob. that information survives:

$$P_R^v=1-\overline{P_R^v}=1-(1-P_1\cdot P_2^{v-1}\cdot P_3^{v-1})^R$$

Results





Replication:

- Increases survival probability
- Requires more storage and power
- Given enough rounds, **Adv always wins**

Encryption

- hides data contents and origin
- Adv can not decrypt
- Adv can't identify data to erase
- Public Key vs. Symmetric key
- Randomized Encryption
  - Distinct random value involved in each encryption operation
  - Given two ciphertexts encrypted under the same key. infeasible to determine whether two corresponding plaintexts equal

Public Key Encryption

- Each node knows sink's public key PK_S
- Data sensed by s_i at round r stored as:

$$E_i^r = E(\underbrace{PK_S, r, s_i, d_i^r, R, etc}_{}.)$$

- Adv can only brute-force guessing plaintext
- Good quality randomness makes plaintext guessing infeasible
- But, where does R come from?
- Dirty little secret…

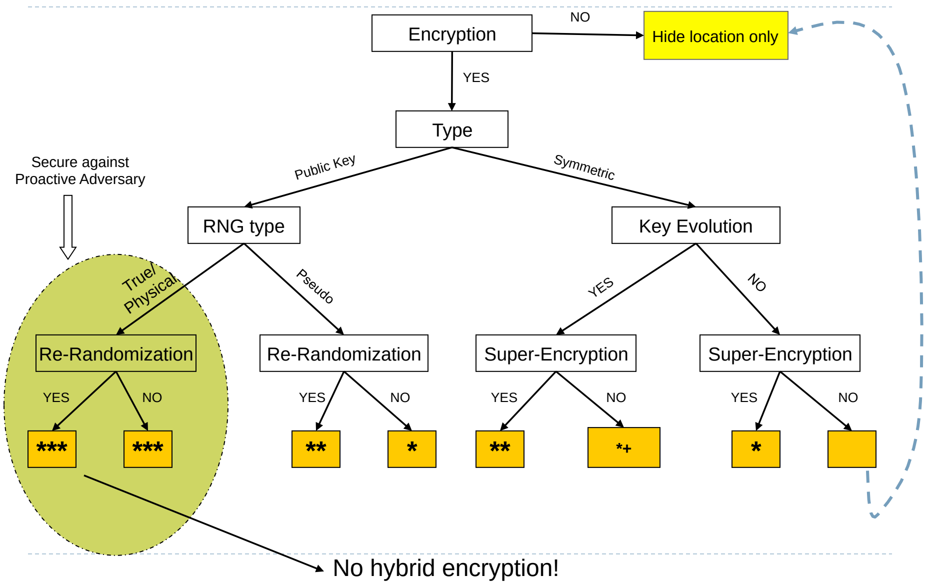
Symmetric Encryption

- Each s_i pre-shares k_i with sink
- Data sensed by s_i at round r stored as:

$$E_i^r = E'(K_i, r, s_i, d_i^r, etc.)$$

- No security…
- Adv breaks in, learns k_i and decrypts E^r_i

“Crypto Flowchart”



Encryption

NO

Hide location only

YES

Type

Public Key

Symmetric

Secure against Proactive Adversary

RNG type

Key Evolution

True/ Physical

Pseudo

YES

NO

Re-Randomization

Re-Randomization

Super-Encryption

Super-Encryption

YES NO YES NO YES NO YES NO

*** *** ** * ** *+ *

No hybrid encryption!

END PART 2

Question:

- How to recover from mobile adversary compromise without per sensor TRNG?

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

IEEE SRDS’08

Motivation

- Curious ADV: wants to learn sensor-collected data in UWSN
- Encryption does not really help
  - Symmetric keys exposed with node compromise
  - With public key encryption, ADV can GUESS plaintext
    - Randomized public key encryption helps but only with a TRNG
    - TRNG neither available nor foreseeable on ultra-cheap sensors
- Can we protect category (1) and (3) data?

Sensor-collected data:

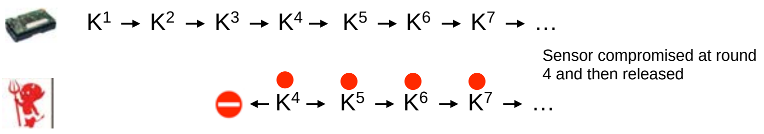
(1) Collected Before Compromise

(2) Collected During Compromise

(3) Collected After compromise (3)

Forward Secrecy

- Even if ADV learns current key, cannot derive PREVIOUS round keys
- Per-round key evolution
  - At each round, key is evolved using a one-way function
    - $K^{r+1}=H(K^r)$
- But, after compromise, ADV can mimic key evolution process



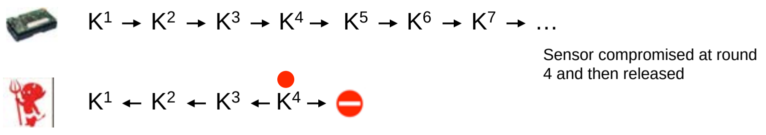
$K^1 \rightarrow K^2 \rightarrow K^3 \rightarrow K^4 \rightarrow K^5 \rightarrow K^6 \rightarrow K^7 \rightarrow \ldots$

⛔ $\leftarrow \overset{●}{K^4} \rightarrow \overset{●}{K^5} \rightarrow \overset{●}{K^6} \rightarrow \overset{●}{K^7} \rightarrow \ldots$

Sensor compromised at round 4 and then released

Backward Secrecy

- Even if ADV learns the current key, cannot derive FUTURE round keys
- Based on assisted per-round key evolution
  - Requires online TTP or secure hw (same as distributed TTP)
- Not suitable for UWSNs
  - Our sink is offline



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

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

Sensor compromised at round 4 and then released

POSH: Main Idea

- Forward secrecy through key evolution
- Backward secrecy via sensor cooperation

  Initial observation:

  – A sensor can securely generate a key unknown to ADV, if it obtains at least one contribution from a non-compromised peer sensor

Network Assumptions 1/2

- **Periodic data collection**
  - Time divided in fixed collection rounds
  - Each (of ***n***) sensors collects single data unit per round
- **Unattended Operation**
  - Itinerant sink periodically visits to collect data
  - ***v*** – maximum # collection rounds between successive sink visits
- **Communication**
  - UWSN always connected
  - Any two sensors can communicate either directly or via peers

Network Assumptions 2/2

- **Storage**
  - Each sensor has enough storage for O(v) data
- **Cryptographic Capabilities**
  - Cryptographic hashing, e.g., SHA-2
  - Symmetric key encryption
    - unique initial secret key shared with sink
  - Pseudo-Random Number Generator (PRNG)
    - unique secret seed shared with sink
- **Re-initialization**
  - During each visit, sink re-initializes ALL sensors (ADV not present):
    - New (or old?) software
    - New secret key
    - New secret seed
    - Empty storage (secure erasure)

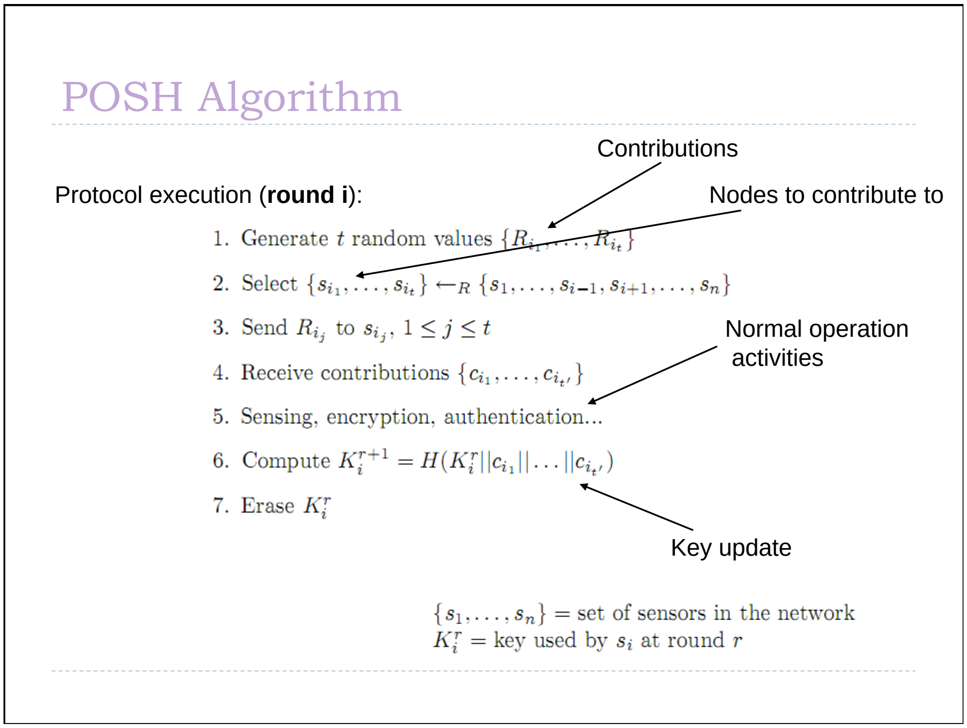
Adversarial model 1/2

- **ADV Goal**
  - learn as many secrets as possible (keys and/or other keying material).

- **ADV Compromise Power**
  - Can compromise at most $0 < k < n/2$ sensors at any round.
  - Reads all storage/memory and listens to all communication of a compromised sensor.
- **ADV Periodic Operation**
  - At the end of each round, picks a subset of up to k
  - At the start of each round, **atomically** releases current sensors and compromises new subset

Adversarial model 2/2

- **Topology Knowledge**
  - Knows the entire topology
- **Minimal Disruption**
  - Does not interfere with sensor behavior
  - Perhaps, in order to remain undetected
- **Defense Awareness**
  - Fully aware of any scheme or algorithm used by the UWSN



POSH Algorithm

Contributions

Protocol execution (**round i**):

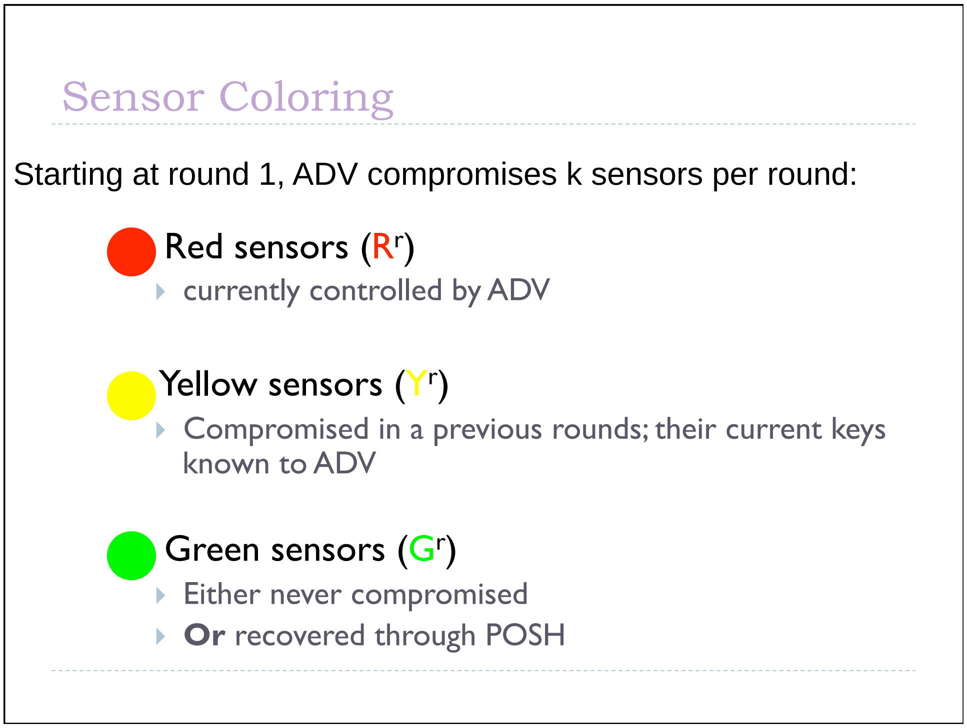
Nodes to contribute to

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

Normal operation activities

Key update

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



Sensor Coloring

Starting at round 1, ADV compromises k sensors per round:

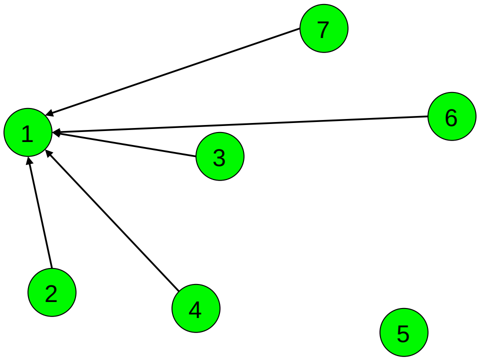
🔴 Red sensors (R^r)
- currently controlled by ADV

🟡 Yellow sensors (Y^r)
- Compromised in a previous rounds; their current keys known to ADV

🟢 Green sensors (G^r)
- Either never compromised
- **Or** recovered through POSH

Example

r = 3



Sensor 1

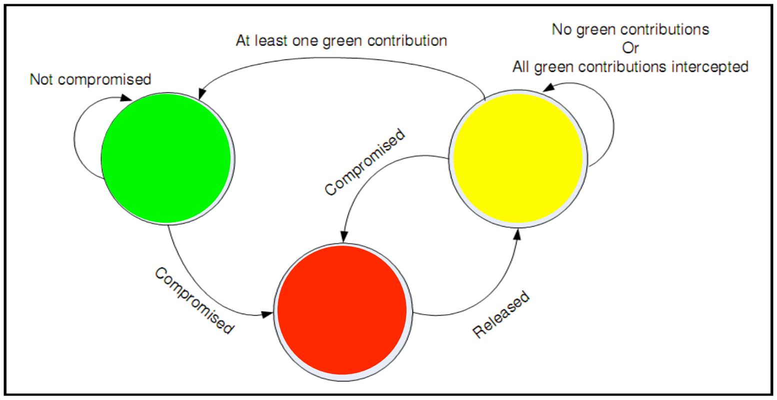
K^1

K^2=H(K^1 || c_3 || c_6)

K^3=H(K^2 || c_2)

K^4=H(K^2 || c_4 || c_7)

Sensor transition diagram



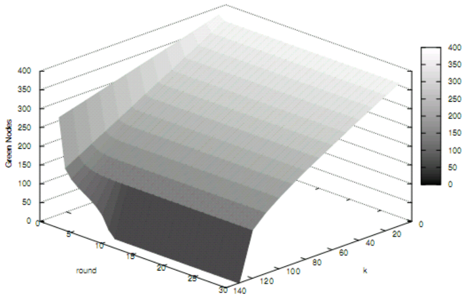
- |R|=k
- ADV's goal – maximize |Y|+|R|
- WSN goal: |G|=n-2k

Two kinds of ADV

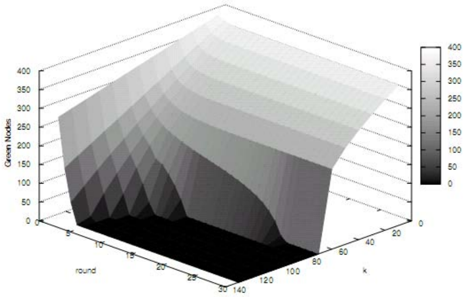
- INF-ADV is always aware of G
  - Unrealistic but very powerful
  - Used as benchmark

- RR-ADV moves through subsets in 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 choose to compromise a yellow sensor

Results: |G| vs INF-ADV



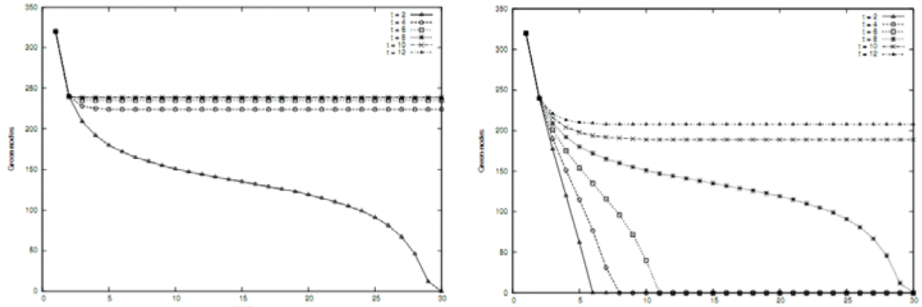
(a) $n = 400, t = 6, p = 0.2$



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

- p = ADV eavesdropping prob.
- t = 6 results in each sensor receiving at least one green 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”

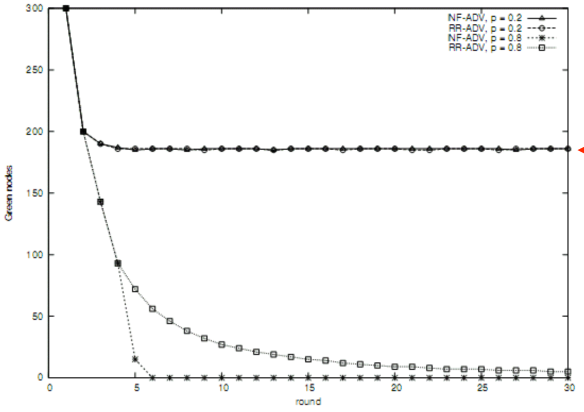


(a) $n = 400, k = 80, p = 0.2$

(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



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

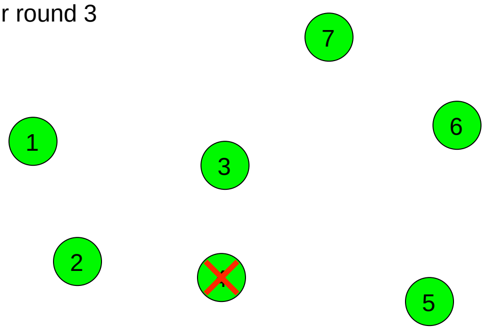
No difference if |G| is close to its optimal value

Dealing w/ real world

- Message delivery failure
  - Sink synchronization
  - Sensor must store IDs of all "contributors" (per round!)
- Sensor failure
  - If sensor fails, its key history cannot be reconstructed
  - Other sensors' secrets might depend on failed one
- Public Key helps here…
  - Encrypt round key with sink's PK
    - Use round key for everything else

Example

Sensor 4 fails after round 3



Sensor 1

K^1

$K^2=H(K^1 || c_3 || c_6)$

$K^3=H(K^2 || c_2)$

$K^4=H(K^2 || c_4 || c_7)$

Sink

K^1

$K^2=H(K^1 || c_3 || c_6)$

$K^3=H(K^2 || c_2)$

$K^4=H(K^2 || ? || c_7)$

K^4 requires sensors 4 and 7 / K^1 is shared

Sensor 1 will have contributed to other peers…

Conclusion

- UWSN security represents new problem domain that calls for new solutions
- No cryptography means no security
- Cryptography helps but not as much as expected
- Cooperation helps a lot
- Role of randomization in UWSN not completely characterized yet

Summary & Directions

- Contributions:
  - New kind of network - UWSN
  - New mobile UWSN adversary
  - Simple approaches simply don't work!
- Lots of interesting problems
- Ongoing and Future work:
  - Mobility?
  - New adversarial models and flavors
    - What if Adv interferes with networking and/or sensing?

Bibliography

- **Intrusion-Resilience in Mobile Unattended WSNs**, IEEE INFOCOM 2010
- **Data Security in Unattended Sensor Networks**, IEEE Transactions on Computers, Vol. 50, No. 11, 2009.
- **New Adversary and New Threats: Security in Unattended Sensor Networks**, IEEE Network, Vol. 23, No. 2, 2009.
- **Playing Hide-and-Seek with a Focused Mobile Adversary in Unattended Sensor Networks**, Ad Hoc Networks, Vol. 7, No. 8, 2009.
- **Collaborative Authentication in Unattended Sensor Networks**, ACM WISEC 2009.
- **DISH: Distributed Self-Healing in Unattended Wireless Sensor Networks**, SSS 2008.
- **POSH: Proactive co-Operative Self-Healing in Unattended Wireless Sensor Networks**, IEEE SRDS 2008.
- **Catch Me (If You Can): Data Survival in Unattended Sensor Networks**, IEEE PERCOM 2008.

Finally… the end!

- Questions?
- Comments?