Data Security in Unattended Wireless Sensor Networks

Gene Tsudik
SCONCE – Secure Computing and Networking Center
UC Irvine
http://sconce.ics.uci.edu

Joint work with:
Roberto Di Pietro
Università di Roma 3
Luigi Mancini
Università di Roma “La Sapienza”
Claudio Soriente
UC Irvine
Angelo Spognardi
Università di Roma “La Sapienza”
Agenda

- Introduction
  - A different kind of WSN
  - New adversarial model (with many flavors)

- Naïve defense strategies

- Cryptography to the rescue

- Related Work

- Conclusions + challenges
A “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
Lots of Prior Work on Sensor Security
Unattended Wireless Sensor Network (UWSN)

- Nodes operate in hostile environment
  - Initial deployment might be ad-hoc
- No ever-present sink
  - Itinerant
- Periodic data sensing (on-demand – N/A, event-driven -- ?)
  - Nodes might retain data for a long time
  - Data might be valuable
- Nodes are left on their own
  - Adversary roams around with impunity
  - **Challenge: Data Survival in UWSNs**
Examples

- WSN deployed in a recalcitrant country to monitor any potential 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.
### UWSN Mobile Adversary

Adv defined by: goal / operation / visibility

<table>
<thead>
<tr>
<th><strong>Goal:</strong></th>
<th><strong>Operation:</strong></th>
<th><strong>Visibility:</strong></th>
<th><strong>Focus:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Search-and-erase</td>
<td>Reactive</td>
<td>Stealthy</td>
<td>General</td>
</tr>
<tr>
<td>Search-and-replace</td>
<td>Proactive</td>
<td>Visible</td>
<td>Targeted</td>
</tr>
<tr>
<td>Curious</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polluter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eraser</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7
# UWSN Mobile Adversary

<table>
<thead>
<tr>
<th>Visibility</th>
<th>Search-and-erase</th>
<th>Search-and-replace</th>
<th>Curious</th>
<th>Polluter</th>
<th>Eraser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stealthy</td>
<td>Proactive Reactive</td>
<td>Proactive Reactive</td>
<td>Proactive</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Visible</td>
<td>Proactive Reactive</td>
<td>N/A</td>
<td>N/A</td>
<td>Proactive Reactive</td>
<td>Proactive Reactive</td>
</tr>
</tbody>
</table>

Adv Goal: Visibility
New kind of Adversary (Adv)

- Well-informed
  - Knows network topology and network defense strategy

- Erratic (seemingly)
  - Unpredictable and possibly untraceable movements

- Mobile
  - Migrates between sets of nodes between sink visits

- Data-centric
  - No interference with sensing or network operation

- Powerful (but not omnipotent)
  - Compromises up to a certain # of nodes
Assumptions

- Scheduled (per round) data sensing/collection
  - Max $v$ rounds between sink visits

- Adv compromises at most $k$ (out of $n$) nodes per round
  - Compromised nodes not necessarily contiguous
  - Reads all storage
  - Listens to all incoming and outgoing communication

- Adv knows which data to target and when it was sensed
  - Receives external signal at collection time
    - Target node identity + collection round
    - Possibly, also target value

- UWSN knows nothing…
  - Equal protection for all data
Does this sound familiar?

Crypto Mobile Adversary

- Ostrovsky & Yung: How to Withstand Mobile Virus Attacks, PODC 1991
- Proactive Cryptography: Decryption and Signatures (e.g., RSA, DSA, Schnorr)
Agenda

- Introduction
  - A different kind of WSN
  - New adversarial model (with many flavors)
- Search-and-Erase Adv: Naïve defense strategies
- Cryptography to the rescue
- Related Work
- Conclusions + challenges
Stealthy Search-and-Erase Adv

IEEE Percom'08, this week in Hong Kong 😊
What we want: whack-a-mole
What if sensors have no crypto capability?

- Cheap sensors
  - No crypto
  - Can only (attempt to) hide data location

- Data Migration strategies
  - Do Nothing
  - Move Once
  - Keep Moving

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

- Adv strategy:
  - Lazy
  - Frantic
  - Smart
## Survival vs. Attack Strategies

<table>
<thead>
<tr>
<th>Survival Strategy</th>
<th>Attack Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LAZY</td>
</tr>
<tr>
<td>DO NOTHING</td>
<td>✗</td>
</tr>
<tr>
<td>MOVE ONCE</td>
<td>✗</td>
</tr>
<tr>
<td>KEEP MOVING</td>
<td>✶</td>
</tr>
</tbody>
</table>
Do Nothing

- Data kept at originating sensor
  - Trivial

- Adversary wins in one round
  - Round 0
    - Learns originating sensor
  - Round 1
    - Corrupts it
    - Deletes target data
Move Once

- Data off-loaded to a random recipient node
  - Kept there for all subsequent rounds

- Adversary wins in at most \( \left\lfloor \frac{n}{k} \right\rfloor \) rounds

- Round 0
  - Learns originating node
  - Data not there anymore

- Round i
  - Move to next set of previously uncompromised nodes

- At most \( \left\lfloor \frac{n}{k} \right\rfloor \) rounds to find and erase it
Algorithm 1: KEEP-MOVING

/* start round 0 */
all nodes sense their values
each node exchanges data with others
0
/* end round 0 */
SET z = min (v, \frac{v}{L})
SET found=FALSE
for ((r = 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 (z found on some s_k in C_r) then
      delete x
      SET found=TRUE
  else
    all nodes sense their values
    each node exchanges data with others
    if (x received by some s_k in C_r) then
      delete x
      SET found=TRUE
  /* end round r */
Keep Moving – Lazy

- Exploit the fact that data is constantly moving among sensors
- Two chances at round 1; one chance each new round
- Prob. data survives v rounds

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

\[ P_1 = \frac{k}{n} + \left(1 - \frac{k}{n}\right) = \left(1 - \frac{k}{n}\right)^2 \]

\[ P_2 = 1 - \frac{k}{n} \]
Keep Moving – Frantic

- Select a new random set of sensors to compromise at each round
- Two chances per round
- Prob. data survives v rounds

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

\[ P_1 = \frac{k}{n} + \left(1 - \frac{k}{n}\right) \frac{k}{n} = \left(1 - \frac{k}{n}\right)^2 \]

\[ P_2 = 1 - \frac{k}{n} \]

\[ P_3 = 1 - \frac{k}{n-k} \]
Keep Moving – Smart

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

Network strategy: Keep-Moving  Settings: n=100, k=10

Survival probability

Rounds

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5 10 15 20 25 30 35 40 45 50

Attacker strategy
Static
Frantic
Smart
Prob. # stored messages do not exceeds a given value $\ell$

- $L_i^r = $ # msg stored on $s_i$ at round $r \rightarrow E[L_i^r] = r$
- From the method of bounded differences, given $\ell > r + \sqrt{rn}$

$$Pr[L_i^r \geq \ell \cup \ldots \cup L_n^r \geq \ell] \leq nPr[L_i^r \geq \ell] \leq e^{-r/2+\ln n}$$
### Overhead 2

<table>
<thead>
<tr>
<th>Scheme</th>
<th>msg × round (r)</th>
<th>msg tot</th>
<th>Queue</th>
<th>msg rec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move-Once</td>
<td>n</td>
<td>v × n</td>
<td>[Pr[\exists s_i \text{ s.t. } L_i^r \geq r + \sqrt{n}r] \leq e^{-r/2+\ln n}]</td>
<td>(O(\ln n))</td>
</tr>
<tr>
<td>Keep-Moving</td>
<td>(r \cdot n)</td>
<td>((r^2/2) \cdot n)</td>
<td>(Pr[\exists s_i \text{ s.t. } L_i^r \geq 2er] \leq 2^{-r+\ln n})</td>
<td>(Pr[\exists s_i \text{ s.t. } M_i^r \geq 2er] \leq 2^{-r+\ln n})</td>
</tr>
</tbody>
</table>

- Prob. # stored messages do not exceeds a given value \(\ell\)
- \(L_i^r = \# \text{ msg stored on } s_i \text{ at round } r\) \(\rightarrow E[L_i^r] = r\)
- From the method of bounded differences, given \(\ell > r + \sqrt{rn}\)
  \[Pr[L_i^r \geq \ell \cup \ldots \cup L_n^r \geq \ell] \leq nPr[L_i^r \geq \ell] \leq e^{-r/2+\ln n}\]
- Variables \(L_i^r\) are independent \(\rightarrow\) Chernoff bound \(\rightarrow Pr[L_i^r \geq \ell] \leq 2^{-r}\) for \(\ell > 2er\)
- \(M_i^r = \# \text{ msg received by } s_i \text{ at round } r\)
  \[Pr[M_i^r \geq \ell \cup \ldots M_n^r \geq \ell] \leq nPr[M_i^r \geq \ell] \leq 2^{-r+\log_2 n}\]
Replication

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

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

$$\overline{P}_R = P_r[X_{1,v} = 0 \land \ldots \land X_{R,v} = 0] = P_r[X_{1,v} = 0]^R = (1 - P_r[X_{1,v} = 1])^R = (1 - P_1 \cdot P_2^{v-1} \cdot P_3^{v-1})^R$$

- Prob. that information survives:

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

Replication of sensed data
- Increases survival probability
- Requires more storage and power
- Given enough rounds, **Adv always wins**
Encryption

- Goal: hide data contents and origin from the adversary
- Adv can not decrypt
- Adv can not identify data to erase
- Public Key vs. Symmetric key

- Randomized Encryption
  - Random values involved in the encryption process
  - Given two ciphertexts encrypted under the same key, it is infeasible to determine whether two corresponding plaintexts are the same
Public Key Encryption

- Each node knows sink’s public key $PK_S$
- $d_{ir}$ -- data sensed by $s_i$ at round $r$ stored as

$$E_i^r = E(PK_S, r, s_i, etc.)$$

- Adv can only try brute-force guessing the plaintext
  - If random data involved in encryption, ciphertext guessing becomes infeasible (i.e., randomized encryption)
Symmetric Encryption

- Each $s_i$ shares $k_i^0$ with the sink
- $d_{i,r}$ -- data sensed by $s_i$ at round $r$ stored as:
  \[ E_r^i = E(k_r^i, d_r^i, r, \text{etc.}) \]

- Forward security
  - per round key evolution:
    \[ k_{r+1}^i = OWF(k_r^i) \]

- Adv cannot compute previous keys
**Crypto Decision Tree**

- **Encryption**
  - **Type**
    - Public Key
    - Symmetric
  - **RNG type**
    - True/Physical
    - Pseudo
  - **Key Evolution**
    - YES
    - NO
  - **Re-Randomization**
    - YES
    - NO
  - **Super-Encryption**
    - YES
    - NO

- **Percom’08**

- **Secure against Proactive Adversary**

- **No hybrid encryption!**
Near-Term Challenges

- How to recover from compromise without PK + TRNG
- What happens if Adv eavesdrops on migrating data?
- Effects of Adv positioning within UWSN topology (to maximize eavesdropping ability)
Related Work

- **Mobile Ad Hoc Networks**
  - Data availability in partitioned MANETs
  - Multi-path routing to improve confidentiality and availability

- **Sensor Networks**
  - Data coding to increase data recovery in presence of disasters
    - [Kamra, et al. 2006]
Conclusion + Future Directions

- Contributions:
  - New kind of network - UWSN
  - New mobile UWSN adversary
  - Simple approaches for data survival simply don’t work!
- Lots of interesting problems
- Ongoing and Future work:
  - Explore the design space of cryptographic techniques
    - Encryption
    - Authentication
  - New adversarial models and flavors
    - What if Adv interferes with networking and/or sensing?
P.S.: A panel at Oakland’05

“Security in Ad-hoc and Sensor Networks”
Panelists: Virgil Gligor, Gene Tsudik, David Wagner

Excerpt from my presentation:
- What research results in sensor net security cannot be applied in more general settings?

(What I really meant is: sensor security is bogus… a mere exercise in contortionism)
The End…

- Questions?
- Comments?
- Complaints?