Addressing Security and Privacy Challenges in Internet of Things

Arsalan Mosenia
Postdoctoral Researcher
Internet of Things

Enabling numerous services over the Internet
Interconnection of heterogenous entities
Over 50B Internet-connected devices by 2020
## Challenges & Research Directions

<table>
<thead>
<tr>
<th>Architectures</th>
<th>Data Analytics</th>
<th>Efficiency</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>New architectures</td>
<td>Huge amount of data</td>
<td>Real-time processing</td>
<td>Security attacks</td>
</tr>
<tr>
<td>Fog/Edge Computing</td>
<td>Heterogeneity</td>
<td>Small battery</td>
<td>Information leakage</td>
</tr>
<tr>
<td>Unused devices</td>
<td>Missing records</td>
<td>Small storage</td>
<td>Security-friendly design</td>
</tr>
</tbody>
</table>

- Real-time processing
- Small battery
- Small storage
- Security attacks
- Information leakage
- Security-friendly design
Security Challenges

Security and privacy
- Existence of insecure in-market products
- Lack of standardization
- Resource constraints
- Unknown threats
- ....

Diagram:
1. Edge devices
2. Communication
3. Edge/Fog
4. Data accumulation
5. Data abstraction
6. Applications
7. Users and centers
Potential Attackers

Attackers:
- Occasional hackers
- Cybercriminals
- Government

Attackers’ Motivations:
- Controlling devices
- Stealing *sensitive* information

IoT-based systems:
- Huge amount of information
- Monitoring/automation
Research Themes

- IoT devices & CPSs
- Learning & Data sciences
- Information Security
Research Themes

IoT & CPS Security

Uncovering Security/Privacy Flaws

- Information Leakage
  - [IEEE TETC, 2016]
  - [IEEE TMSCS, 2017]
  - [Survey, IEEE TMSCS, 2017]

- Security Vulnerabilities
  - [IEEE TETC, 2017]
  - [ATC USENIX, 2018]

Development of Security-friendly Systems

- Wearables & Implants
  - [IEEE TMSCS, 2015]
  - [UbiComp, 2018]

- Smart Vehicles
  - [IEEE TC, 2017]
  - [UbiComp, 2018]

- Underlying Networks
  - [IEEE TMSCS, 2017]
  - [USENIX Sec, 2018]
  - [FWC, 2018]
  - [Survey, ACM EDA, 2017]
OpenFog Consortium

Founders
PRINCETON UNIVERSITY, intel, cisco, Microsoft, arm, DELL

Contributing Members
FOXCONN, HITACHI, GE, SAKURA, internet, ZTE

Affiliations
Barcelona Supercomputing Center, ETSI, IoT Acceleration Consortium, IEEE ComSoc, IITI, IMEC

We define security standards for Fog/Edge Computing
[2 position papers, Fog World Congress, 2017]

61 members strong, headquartered in 17 countries as of January 2018
Outline

PinMe: Tracking a User Around the World

ProCMotive: Bringing Programmability and Connectivity to Vehicles
IoT & CPS Security

Uncovering Security/Privacy Flaws
- Information Leakage
- Security Vulnerabilities

Development of Security-friendly Systems
- Wearables & Implants
- Smart Vehicles
- Underlying Networks
Location Privacy

Attacks against location privacy lead to:
- advertisement, spams, or scams
- disclosure of personal activities
- ...

Location privacy: determining *when, how, and to what extent* location data are shared
Prior Attacks on Location Privacy

Fundamental limitations of previous attacks:

- Substantial prior knowledge of the path
- An attack-specific training dataset
- Very limited accuracy, e.g., less than 45%

The extent of location-related information that can be inferred from presumably non-critical data was not well-understood!

PowerSpy (GPS is off) [Michalevsky et al.]
Fundamental Challenges

A realistic privacy attack:

- Minimal prior knowledge
- No attack-specific training dataset
- High accuracy
- Different activities
- Robustness

PinMe may offer a promising navigation system for autonomous vehicles
Sources of Information

Permission-free data

- GPS
- Device's IP
- Network status
- Timezone
- Acceleration
- Air pressure
- Heading

Departures

<table>
<thead>
<tr>
<th>Time</th>
<th>Flight</th>
<th>Departure</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00</td>
<td>OD 1961</td>
<td></td>
</tr>
<tr>
<td>12:15</td>
<td>PH 0034</td>
<td></td>
</tr>
<tr>
<td>12:20</td>
<td>T3 0529</td>
<td></td>
</tr>
<tr>
<td>12:30</td>
<td>PH 2415</td>
<td></td>
</tr>
<tr>
<td>12:45</td>
<td>GI 1872</td>
<td></td>
</tr>
<tr>
<td>12:55</td>
<td>T3 0944</td>
<td></td>
</tr>
<tr>
<td>13:00</td>
<td>SF 2778</td>
<td></td>
</tr>
<tr>
<td>13:15</td>
<td>OD 0061</td>
<td></td>
</tr>
<tr>
<td>13:30</td>
<td>BK 1532</td>
<td></td>
</tr>
</tbody>
</table>
Step 1: Dynamic Partitioning & Activity Classification

What if the user shakes the phone? **Merging**

Activity classification (4 SVMs):
- Air pressure
- Acceleration
- Heading (compass)
Step 2: Tracking the Vehicle

Update the tree

Estimate the elevation (E)

Find a turn 120 > \( \Delta H > 60 \)

Construct a navigational tree

[\( \Delta H, E \)]

Show routes

IP1
IP2
IP3
IP3
IP3
IP4
IP4

C
C
W
W
W
C
C

Air pressure

Heading

\[ E_{\text{turn}} = E_{\text{station}} + \frac{T}{C} \ln \left( \frac{P_{\text{turn}}}{P_{\text{station}}} \right) \]
Real-world Evaluation

1. Three smartphone: Galaxy S4 i9500, iPhone 6S, and iPhone 6

2. Two datasets:
   - Set #1: 405 data chunks collected during different activities (271 chunks for driving)
   - Set #2: 3 data streams collected by 3 users (Mazda 3, Mazda CX7, Toyota Camry)
Results: Tracking the Vehicle

The number of possible routes drops rapidly!
Results: End-to-end Evaluation

The accuracy of PinMe is comparable to GPS

Trajectories of three different users. Starting from the left and moving to right: (a) Princeton [Galaxy S4 i9500], (b) Princeton [iPhone 6], and (c) Baltimore [iPhone 6S]
## Comparison

<table>
<thead>
<tr>
<th>Tracking mechanism</th>
<th>#Activity</th>
<th>Prior info.</th>
<th>Training</th>
<th>OS</th>
<th>Sampling freq.</th>
<th>Device/Vehicle dependence</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACComplice Han et. Al, 2012</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Android</td>
<td>30 Hz</td>
<td>Y</td>
<td>10%*</td>
</tr>
<tr>
<td>PowerSpy Michalevsky et al., 2015</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Android</td>
<td>N/A</td>
<td>Y</td>
<td>45%</td>
</tr>
<tr>
<td>Narian et al., 2016</td>
<td>1</td>
<td>N</td>
<td>N</td>
<td>Android</td>
<td>20-100</td>
<td>Y</td>
<td>10%*</td>
</tr>
<tr>
<td>PinMe</td>
<td>4</td>
<td>N</td>
<td>N</td>
<td>Android</td>
<td>5 Hz</td>
<td>N</td>
<td>100%</td>
</tr>
</tbody>
</table>
Summary and Future Work

PinMe:
- sheds light on information leakage from seemingly-benign data
- offers a promising alternative to GPS

We:
- are performing a large-scale study
- started conversations with companies

U.S. Patent Pending
Extensive media coverage (e.g., Schneier on Security & Android Authority)
IoT & CPS Security

Uncovering Security/Privacy Flaws
- Information Leakage
- Security Vulnerabilities

Development of Security-friendly Systems
- Wearables
- Implants
- Smart Vehicles
- Underlying networks
State-of-the-art Vehicles

Stats:
- Over 1B vehicles, 78M vehicles sold in 2017
- Average age of vehicles > 12 years
- Most of them do not support connectivity/programmability
Transmitters

Shortcomings:
1. Unavailability of service when wireless is lost
2. Lack of programmability
3. Significant cellular data usage
4. Intolerable response time
5. Security
6. Privacy

Product Recall
New Vehicular Apps

Enabling data-dominant, latency-sensitive, mission-critical, and privacy-sensitive applications
Architectural Overview

Key observations:
- Direct access to critical components
- Vulnerable congestion control
- No access control
Design Goals

**Connectivity**
- Vehicle-to-Cloud
- Vehicle-to-phone
- Vehicle-to-Vehicle

**Security**
- Access control
- Virtualization (containers)

**Privacy**
- Data manipulation
- Minimal transmission

**Programmability**
- Customized Apps
- Low response time

**Cost**
- Minimal transmission
Vehicular Add-on Middleware

Security Measures:
- App Isolation: Containers
- Congestion Control & Probing
- Context-aware Access Control
- Remote Update

Flask-based Web Server

OS: Raspbian

CARWare:
- Update Management
- Application Management
- Access Control
- Port Management
- Data Collection
Data Collection

Enabling data collection from

- Built-in sensors
  20-40 sensors, e.g., speed, RPM
- Add-on modules:
  - GPS receiver
  - Camera
  - BLE-based Sensor Tag

R = [{“appID”: “<ID>”, “appToken”: <Token>, “requestType”: “dataCollection”}, {“source”: “vehicle”, “type”: “vehicle_speed”}]

Response = requests.post(webserver_url, R, headers={'Content-type':'application/json'})

......
Data Collection (Cont.)

\[ R = [\{\text{"appID"}: \"<ID>\", \text{"appToken"}: <Token>, \text{"requestType"}: \"dataCollection\"\}, \{\text{"source"}: \"vehicle\", \text{"type"}: \"vehicle_speed\"\} ] \]

Access Control: Policy Enforcement

getSpeed()
Access Control

Policy types:

- Strict
- Context-aware (over 10 contexts)
  1. Location-based
  2. Operational (e.g., idle/moving)
     - Example: Only send controlling commands when the vehicles is not moving!
  3. Situational (e.g., accident)
Access Control (Cont.)

Policy File

- `{“source”: “vehicle”, “type”: “vehicle_speed”, “policyType”: “strict”, “access”: “always” }
- `{“source”: “GPS”, “type”: “location”, “policyType”: “situational”, “situation”: “accident” }`
Port Management

Public functions:

- Dongle isolation
- Congestion control (rate adjustment)
- Probing
Case Study I: Insurance Monitor

Usage-based insurance plans offer very low rates!

However, their acceptance is limited:

- **Security**
  - Injecting commands [Savage et al., 2015]
  - Denial-of-service attacks

- **Privacy**
  - Reading the vehicle's private data
  - Tracking the vehicle [Gao et al., 2014]
Case Study I: Insurance Monitor

Security:
- Access control
  - Dongle can only read speed
- Port management
  - Behavioral analysis
    - Statistical analysis
    - Learning the profile

Privacy:
- Port management
  - Data manipulation
    - Example: Noise addition
Results: Prevention of Command Injection

- Legitimate requests:
  - 100 requests (querying speed data) with the frequency of 1 → forwards all requests to the vehicle

- Illegitimate requests:
  - 100 attempts to query other data → requests are dropped
  - 100 queries with a high frequency → puts requests in a queue
Case Study II: Experimental Results (Cont.)

Enhancing privacy: (i) shuffling, (iii) shuffling & rounding, (iii) noise addition

Noise addition: $V_i = V_i + Z_i$, where $Z_i$ drawn from a uniform distribution with the range of $R$

Utility = No. of Speed Violations (Speed >30mph)
Case Study II: Amber Response

Stats:

43 children have been recovered every year
800,000 children are abducted in the U.S. every year

A more effective approach is highly needed
Case Study II: Amber Response (Cont.)

Three implementations:
- Cloud-based: On-cloud plate recognition
- SmartCore-based: Local plate recognition
- Hybrid: Plate area detection and color detection on SmartCore

<table>
<thead>
<tr>
<th>#</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black</td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
</tr>
</tbody>
</table>

Few sensitive images:
- Enhanced privacy
- Reduced Costs
- Similar accuracy & Performance
ProCMotive can revolutionize vehicular industry

UbiComp 2018
U.S. Provisional Patent
Innovation Award (2017), IP Accelerator Award (2018)
Thank you!