University of Central Florida

AnonySense: Privacy-Aware People-Centric Sensing

Leith Tussing

EEL 6788

Spring 2010

Overview

- Introduction
- Security Risks
- Architecture
- Protocols
- Evaluation
- Future Work
- Resources

Introduction

- Data collection through personal mobile devices is quickly becoming a viable option for mass data gathering.
- Mobile devices are becoming more feature rich with each iterative generation. What used to be a multi thousand dollar bulky unit is now a sub $100 cell phone.
- As the micro computer replaced the mainframe systems of yore, mobile devices are rapidly replacing the desktop and laptop machines of today.

Introduction (continued)

- Mobile devices will be the super computers of tomorrow. Time slicing tasks across billions of free floating devices. As of the end of 2009 60% (4 billion) of the world population use cell phones.
- As mobile devices are more widely used for data collection the larger the risk of privacy invasion of user's data and personal information.
- As you collect more accurate data you end up with less privacy for the individual collecting the data. Inversely increasing the privacy results in less accurate data for those requesting information.

Introduction (continued)

- More researchers are turning to opportunistic sensing to gather results where there is no fixed sensing or the inability to add it.

- Even though it’s a best-effort service it offers a low cost and large mobile infrastructure.

- Issues with opportunistic sensing
  1. Heterogeneous & unpredictable collection of devices
  2. Interface via autonomous WAPs & public Internet
  3. Poses new and mostly untackled security risks.

Security Risks

- Location/Time based user identifications attacks
- Rogue AP hosting & spoofing
- Mobile device/sensor/software tampering
- Maliciously crafted tasks
- Server spoofing
- Packet sniffing
- Data spoofing/manipulation

AnonySense Architecture

- The author's of this paper proposed and implemented a prototype of a system they called AnonySense, "a privacy-aware architecture for realizing pervasive applications based on collaborative, opportunistic sensing by personal mobile devices."
- An application independent infrastructure for handling anonymous tasking and reporting.
- Built on the idea of minimal trust and task separation to minimize risk.

Architecture: Mobile Nodes

- Mobile Nodes (MN)
  - These are the physical devices carried by people or attached to objects.
  - MNs communicate with the TS & RS via WAPs. WAPs and their providers are untrusted entities. All communications are done over SSL encrypted channels to prevent compromised WAPs from viewing or changing data. MNs use MAC rotation to prevent WAPs from tracking their activity.
  - MNs sign all traffic with short-group keys to prevent unique key signings being used to track traffic back to a single MN.
  - MNs only trust tasks from the TS that match signed certificates from the RS.

Short-Group Signature

- A Group Signature scheme is a way for allowing an anonymous member of a group to sign a message on behalf of the group.

- The key aspects of a GS scheme are; Soundness and Completeness, Unforgeable, Anonymity, Traceability, Unlinkability, No Framing, and Unforgeable tracing verification.

- A Short-Group Signature is one with a signature length less than 200 bytes compared to an RSA based Group Signature which can be 2 kb (2048 bytes) or larger.

Architecture: Registration Authority

- Registration Authority (RA)
  - Registers the MNs that wish to participate.
    - Verifies interpreter is properly installed and sensors are calibrated.
    - Verifies MN attributes.
    - Installs private “short-group key” for signing reports anonymously.
  - Issues certificates to the TS and RS. This gives the ability to verify the authenticity of the services.
  - The RA trusts nothing about any other component.
  - The RA validates the quantity and types of MNs to prevent targeted tasks from being run that may expose specific MNs. A group must consist of a minimum number of MNs before a task can be run against them to protect the MNs.

Architecture: Task Service

- Task Service (TS)
  - Accepts tasks from applications using AnonySense, performs consistency checking, and distributes tasks to MNs as they request new ones.
  - The TS does not trust calling applications, therefore every task is validated for syntax correctness by the TS and tagged with a unique identifier.

Architecture: Report Service

- Report Service (RS)
  - Stores the received reports from MNs and provides data to queries from applications.
  - The RS only trusts tasks that the RA label as valid for dissipation to MNs.
  - Like the TS the RS also does not trust calling applications and requires identification verification prior to permitting it to get report data.

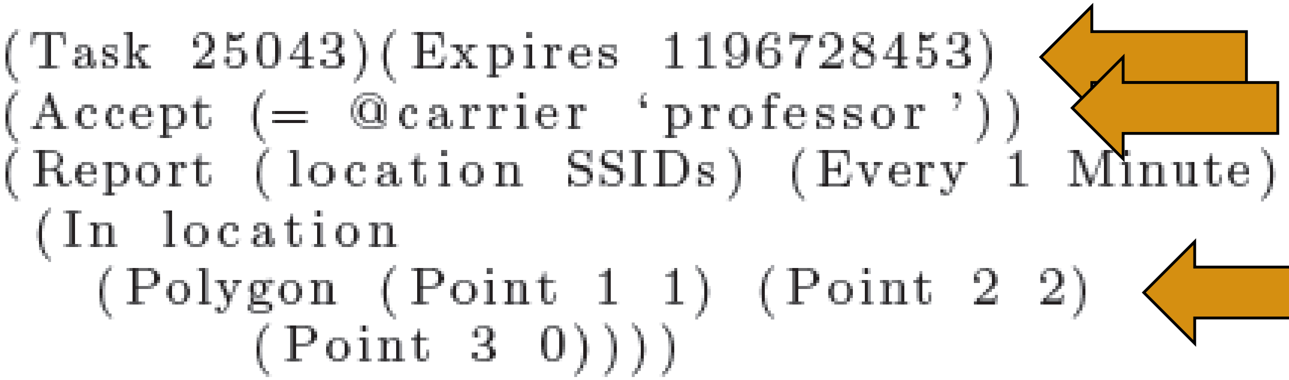
Architecture: MIX Network

- Mix Network (MIX)
  - Anonymizing remailing SMTP servers that utilize encrypted communication channels. The software used is Mixmaster.
  - Gathers report emails from the MNs and holds on to them until a minimum threshold of messages are reached. Once this threshold is reached it passes them along to the RS.
  - Multiple MIX servers are used and MNs randomize which ones they send various messages to.
  - By limiting the span that reports come in to the RS and the MIX servers they come from you anonymize the MNs that send the data.
  - Introduces an increased level of latency though for increased privacy.

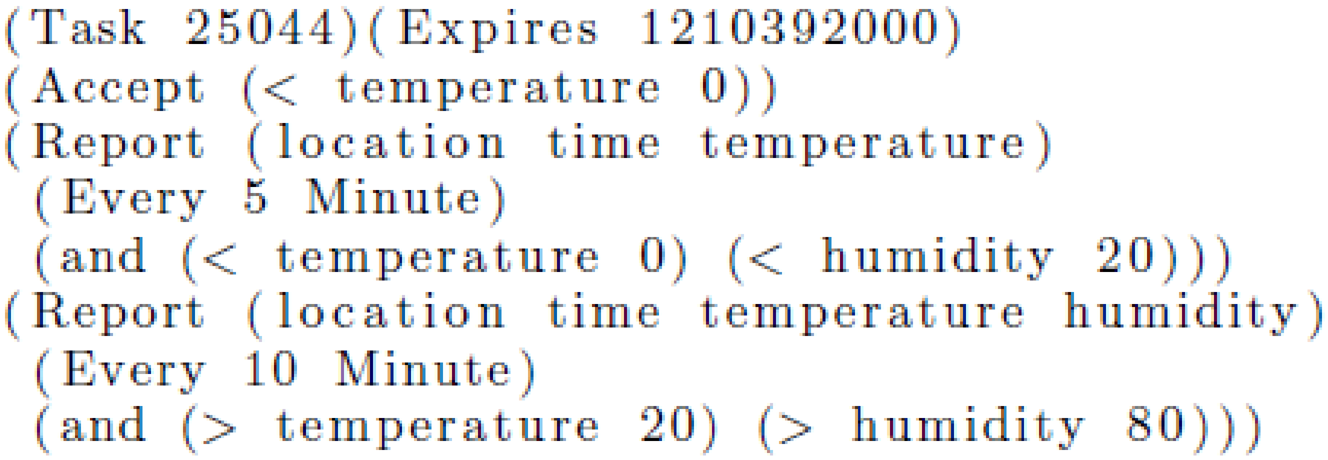
Architecture: AnonyTL

- To simplify tasks and enhance security they defined their own language called AnonyTL.
- AnonyTL specifies a tasks behavior in terms of acceptance conditions, report statements, and termination conditions.
- No code is ever actual transmitted, only instructions. This keeps data packets small and secure. This prevents the system from ever interacting with functions outside of its definition or allowing data to be injected.
- AnonyTL uses a Lisp-like syntax.

AnonyTL



```
(Task 25043)(Expires 1196728453)
(Accept (= @carrier ‘professor’))
(Report (location SSIDs) (Every 1 Minute)
 (In location
   (Polygon (Point 1 1) (Point 2 2)
            (Point 3 0))))
```

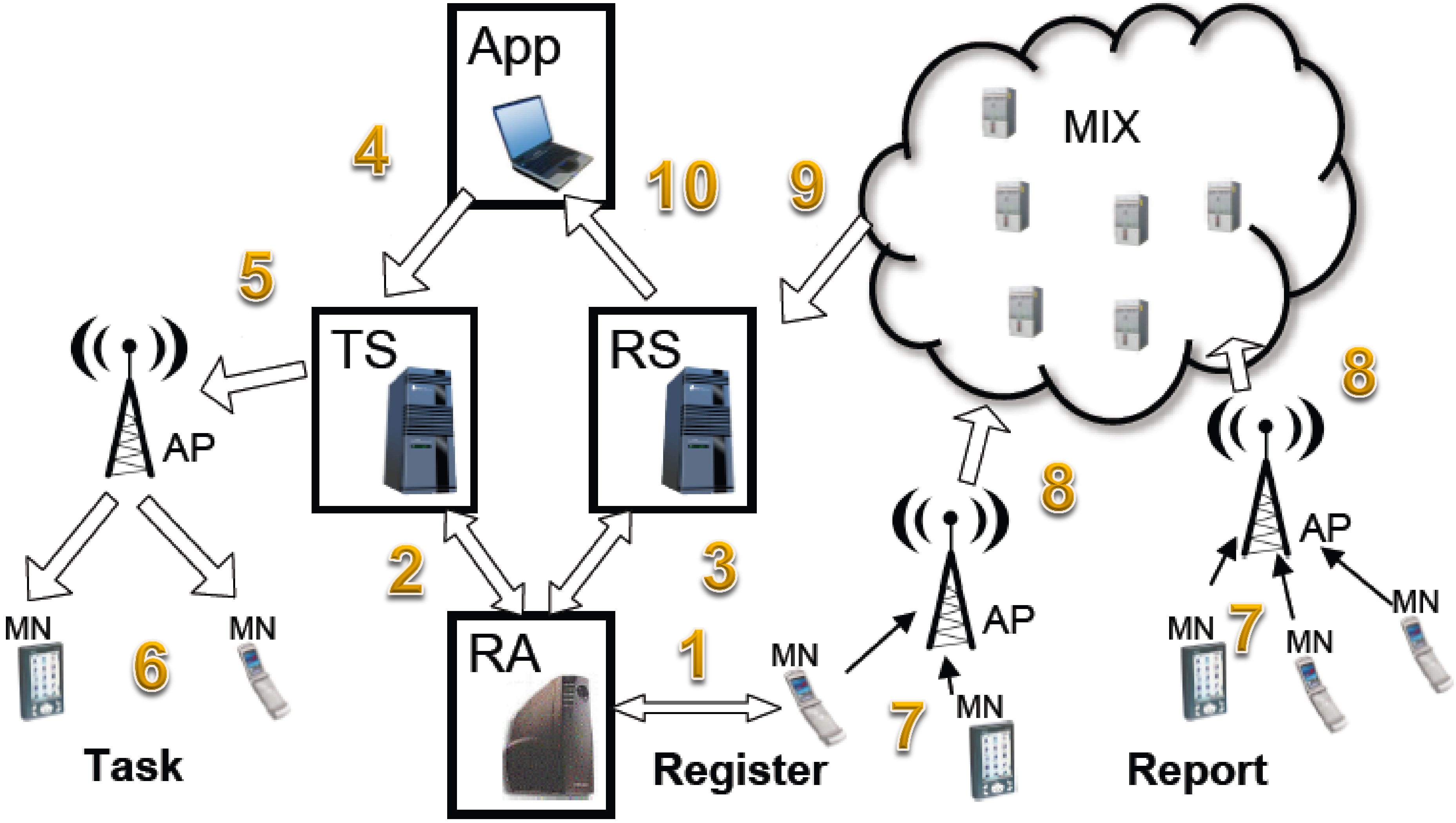


```
(Task 25044)(Expires 1210392000)
(Accept (< temperature 0))
(Report (location time temperature)
 (Every 5 Minute)
 (and (< temperature 0) (< humidity 20)))
(Report (location time temperature humidity)
 (Every 10 Minute)
 (and (> temperature 20) (> humidity 80)))
```

Protocols: Tasking Protocol

- Tasking Protocol
  - All steps use at a minimum SSL encrypted connections.
  - TS to RA communication is also done via mutual authentication which involved SSL certificate handshakes to validate each other as who they say they are.
  - MNs prove to the TS they are a valid MN through group key validation.
- Reporting Protocol
  - The MN signs each report with the group key and then encrypts it using the public key from the RS server.
  - The RS decrypts the message using the private key and then validates the group key.

AnonySense Data Flow



Evaluation: Applications

- RogueFinder – GPS + Wi-Fi
  - Finds WAPs and reports them to have reports run to determine rogue WAPs.
- ObjectFinder – GPS + BT
  - Finds BT exposed MAC addresses and reports them to have reports run to determine the location of a lost BT item.
- QuietFinder – GPS + Mic
  - Measures sound levels and reports them to have reports run to determine quiet places on campus.

Evaluation: Prototype Environment

- All services are written in Ruby programming language.
- The TS and RA implemented using Camping, which is a micro-framework HTTP server, and Mongrel to serve SSL.
- SQLite3 is the database subsystem for all services.
- The RA, TS, RS, MIX, and application were run from a single Linux desktop.
- The MN was a Nokia N800 MID which runs a version of Linux. They mention it runs on the iPhone as well but later define only jailbroken phones since Apple restricts the APIs they need to call, and to run as a service.

Evaluation: Nokia N800 (MID)

Evaluation: Nokia N800 (MID)

- Mobile Internet Device (precursor to the Tablet PC HP Slate, iPad, etc.)
- Maemo Linux (Debian based) developed by Nokia but as of this year (2010) will merge with Moblin, Intel's **Mob**ile **Lin**ux designed for MIDs, Netbooks, & Nettops
- 330 MHz TI ARM CPU with 128 MB of RAM
  - iPhone 3GS – 600 MHz ARM with 256 MB of RAM
  - Nexus One – 1 GHz ARM with 512 MB of RAM
- 802.11B & Bluetooth 2.0

Evaluation: Power Draw

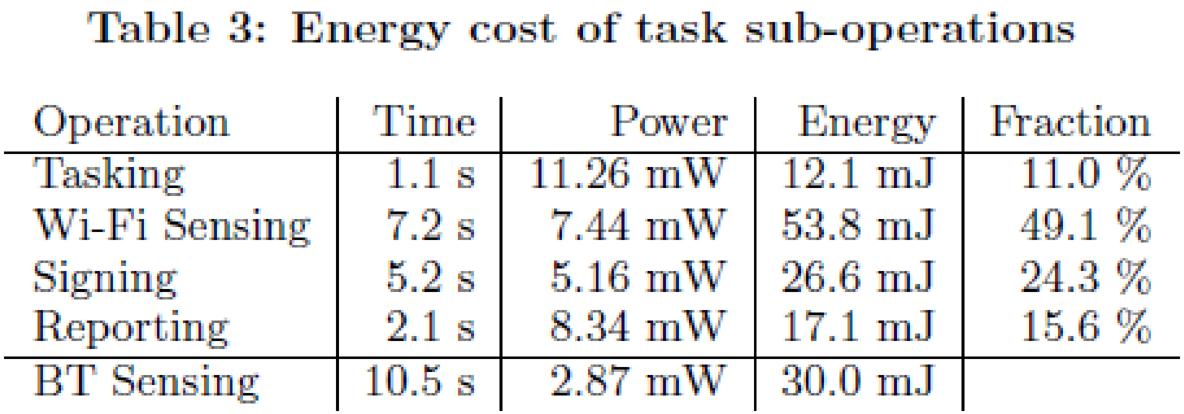


Table 3: Energy cost of task sub-operations

Operation	Time	Power	Energy	Fraction
Tasking	1.1 s	11.26 mW	12.1 mJ	11.0 %
Wi-Fi Sensing	7.2 s	7.44 mW	53.8 mJ	49.1 %
Signing	5.2 s	5.16 mW	26.6 mJ	24.3 %
Reporting	2.1 s	8.34 mW	17.1 mJ	15.6 %
BT Sensing	10.5 s	2.87 mW	30.0 mJ	

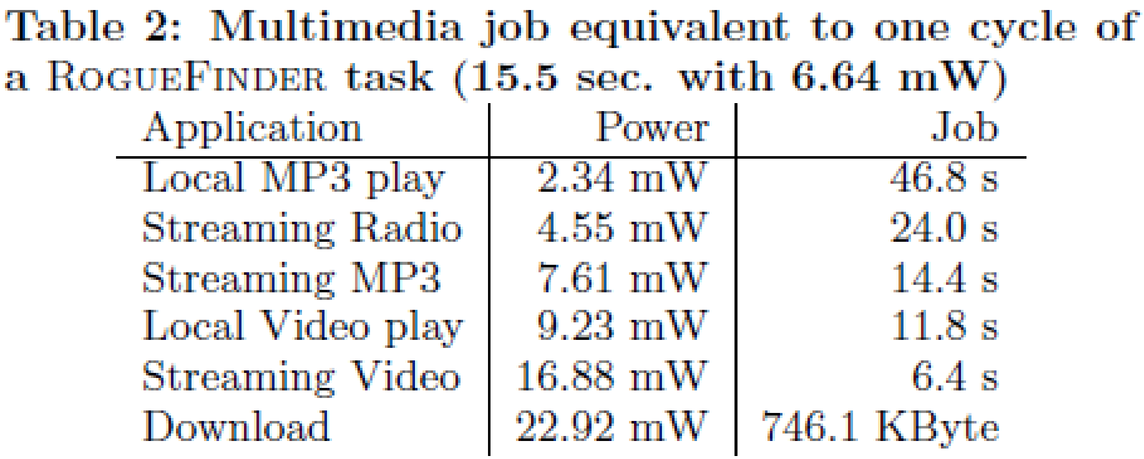


Table 2: Multimedia job equivalent to one cycle of a RogueFinder task (15.5 sec. with 6.64 mW)

Application	Power	Job
Local MP3 play	2.34 mW	46.8 s
Streaming Radio	4.55 mW	24.0 s
Streaming MP3	7.61 mW	14.4 s
Local Video play	9.23 mW	11.8 s
Streaming Video	16.88 mW	6.4 s
Download	22.92 mW	746.1 KByte

Evaluation: Data

- Over the life of a task there was a total of 32.3 KB of data transferred by the MN.

- After analyzing the power draw for a series of tasks they performed long term battery usage tests. Using a once a minute RogueFinder analysis they determined that it reduced the N800’s battery life by 5.26% dropping the typical 285 minute battery life to 270 minutes.

Future Work

- Require applications to authenticate as well.
- TPM (Trusted Platform Module) integration.
- Implement MAC rotating.
- Implement task randomization.

My Thoughts

- - Mishmash of components
- - Components chosen are known to be vulnerable

- + Framework is extremely well thought out
- + Multi layer approach to every aspect of security

References

- AnonySense: Privacy-Aware People-Centric Sensing
  - Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos
- Short Group Signatures
  - Dan Boneh, Xavier Boyen, and Hovav Shacham
- Wikipedia, the free encyclopedia
  - http://en.wikipedia.org
- K-Anonymity: A Model for Protecting Privacy
  - Latanya Sweeney