

Secure Hospital Contamination Prevention

202C Final Presentation
Fall 2015

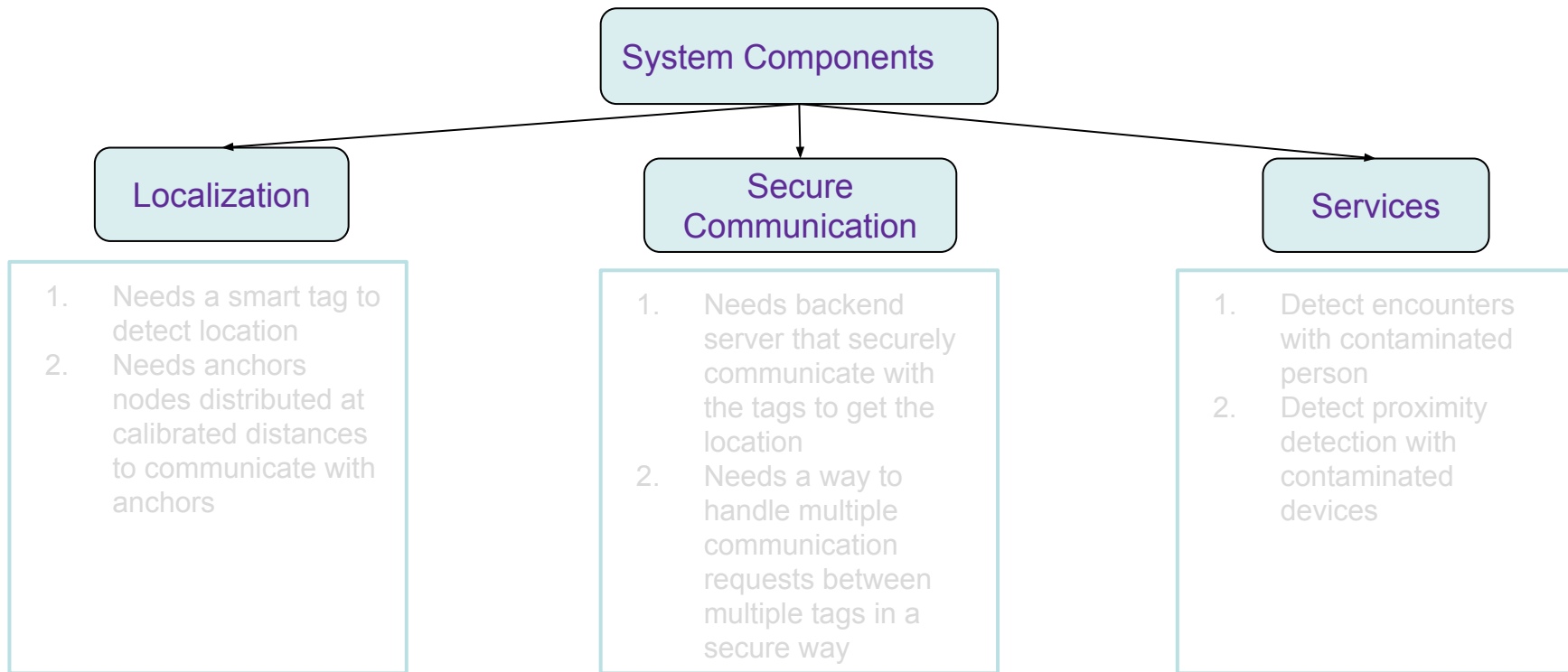
Anthony Nguyen

Pranjal Rastogi

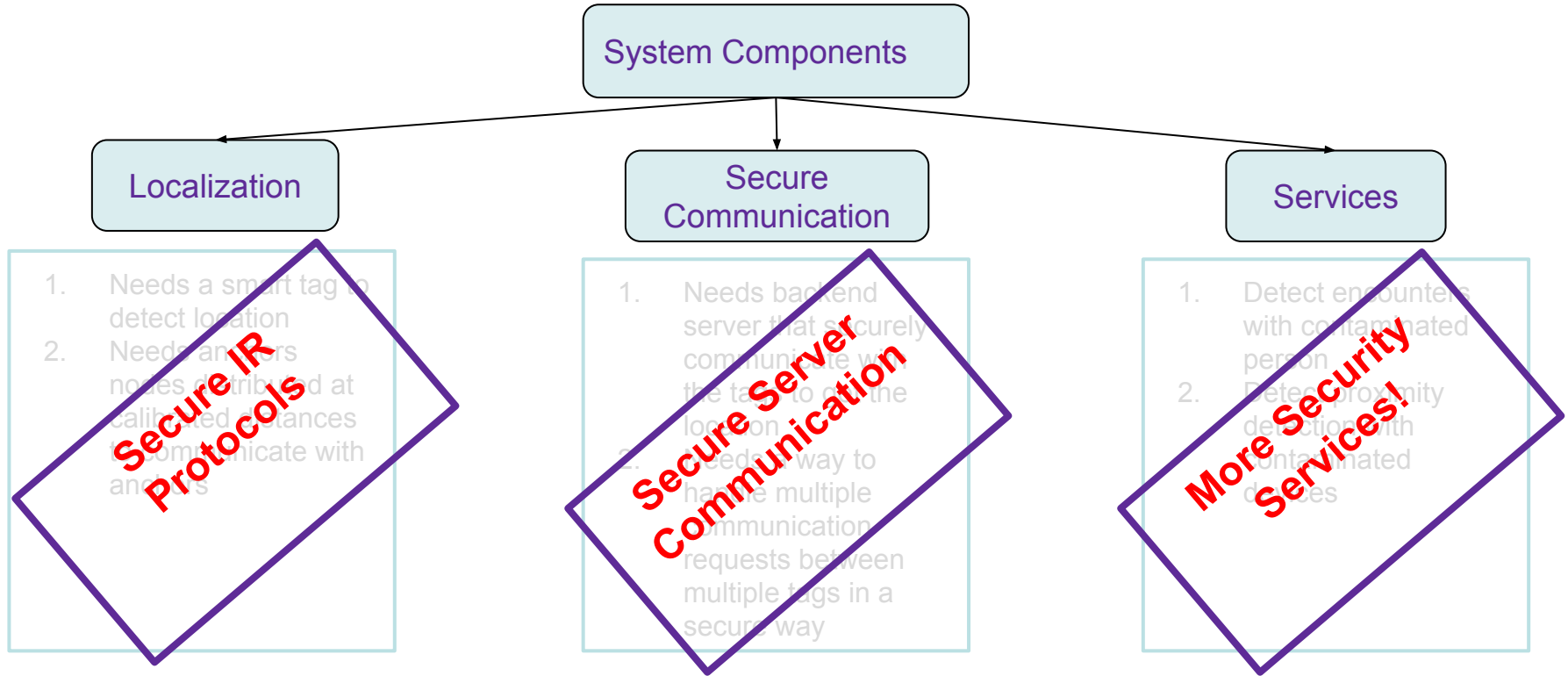
Raymond Andrade

Salma Elmalaki

Recap!



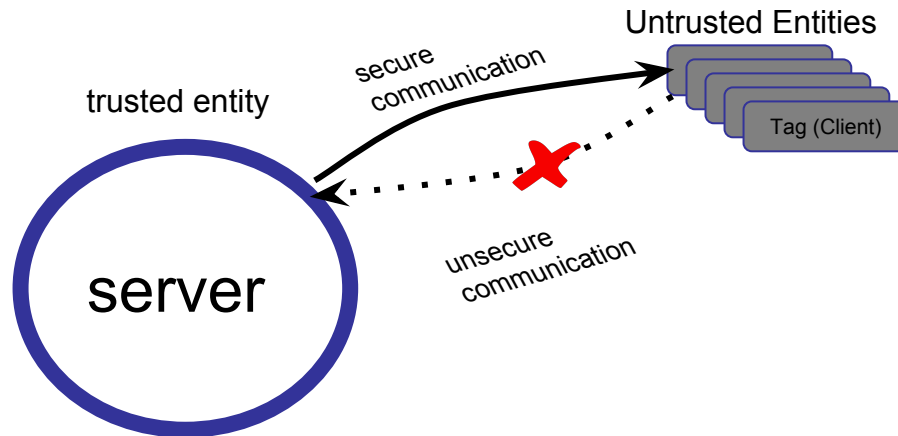
Dive more into security!



Threat Model and System Guarantees

Threat Model and Assumptions:

1. We assume that the server is a trusted entity
2. Server has root access to all the Tags available
3. Server has root access to all the Anchors available



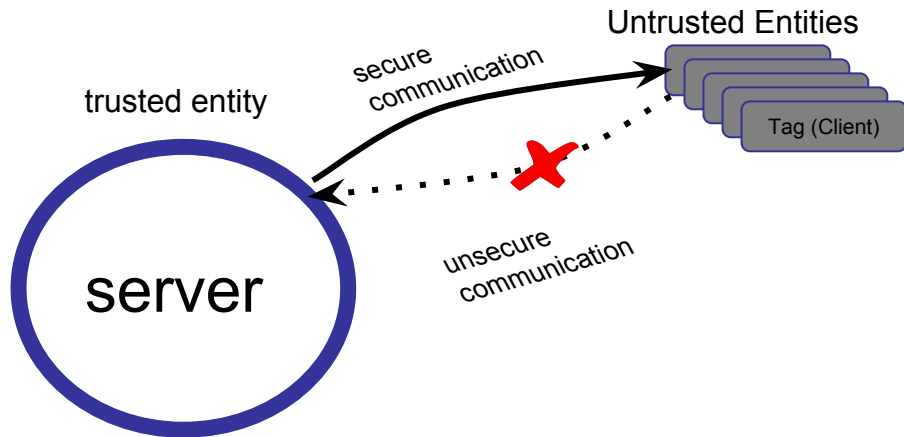
Threat Model and System Guarantees

Threat Model and Assumptions:

1. We assume that the server is a trusted entity
2. Server has root access to all the Tags available
3. Server has root access to all the Anchors available

System Guarantees:

1. Server is the only entity that can keep track of the information and the contamination information
2. Tags have no access to the server



Threat Model and System Guarantees

Threat Model and Assumptions:

1. We assume that the server is a trusted entity
2. Server has root access to all the Tags available
3. Server has root access to all the Anchors available

System Guarantees:

1. Server is the only entity that can keep track of the information and the contamination information
2. Tags have no access to the server

Goals:

1. Accuracy
2. Privacy
3. Integrity

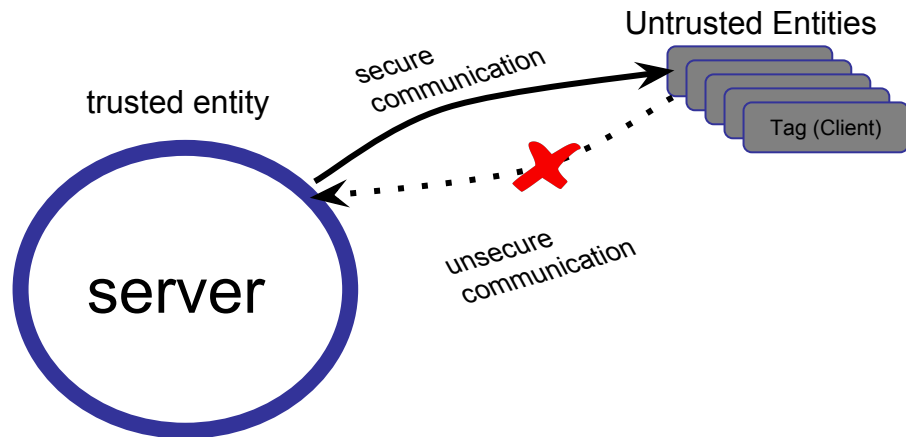


Table of Contents

1. **Server Communication**
2. **IR Communication Security**
3. **Technical Details**
4. **Attacks and Countermeasures**
5. **Future Work**
6. **Demo**

Table of Contents

1. Server Communication

- a. Architecture Models and Design Approach
- b. Main protocol
- c. Control protocol
- d. Service protocol - (Protected Zone, Contamination, and Tracking)

2. IR Communication Security

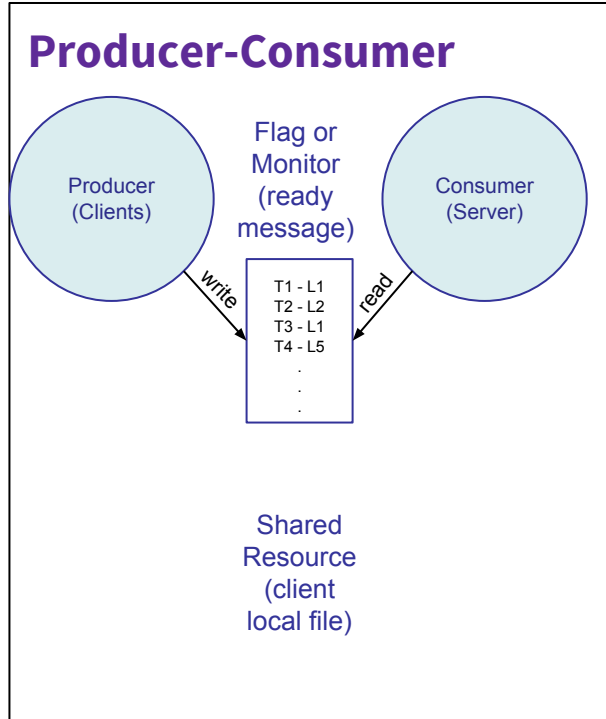
3. Technical Details

4. Attacks and Countermeasures

5. Future Work

6. Demo

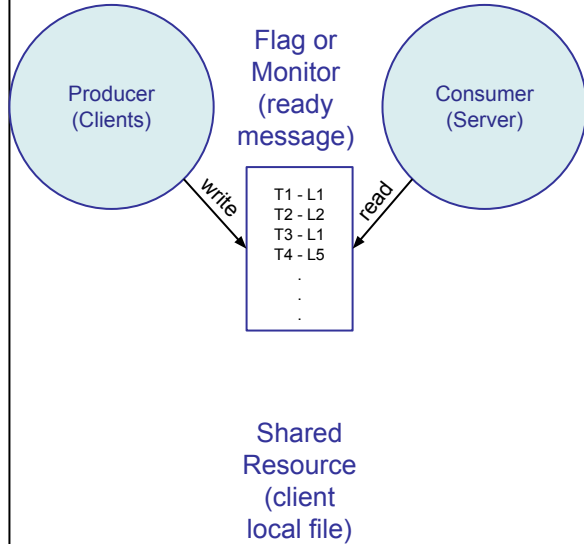
Communication Design Architecture



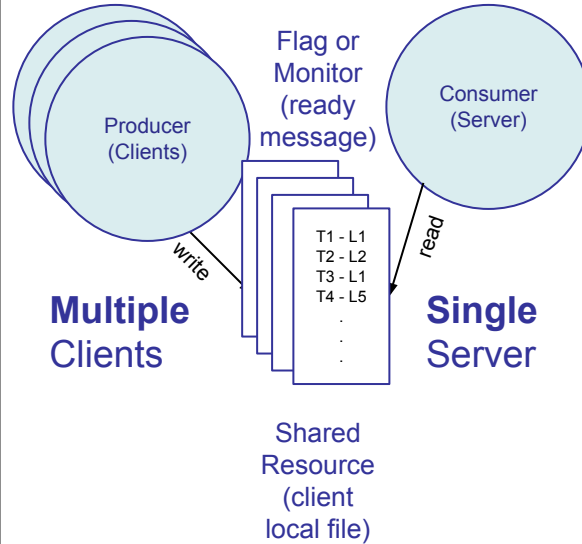
Architecture Model:

1- multiple clients - single server

Producer-Consumer



Architecture Model



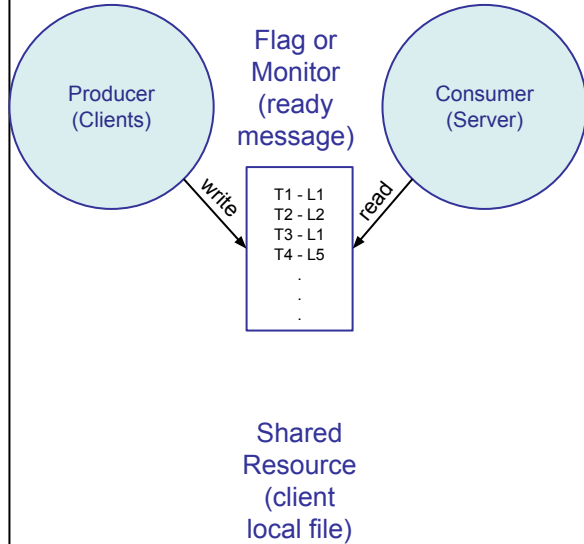
Design Approach

1. Threaded dispatch
asynchronous Server

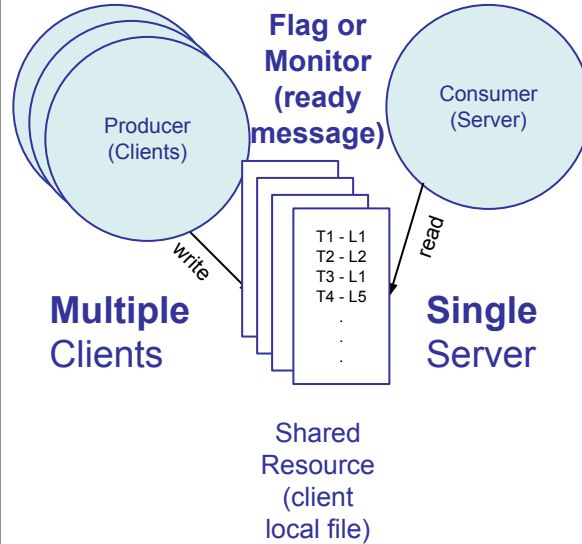
Architecture Model:

2- integrity of information - no loss of data

Producer-Consumer



Architecture Model

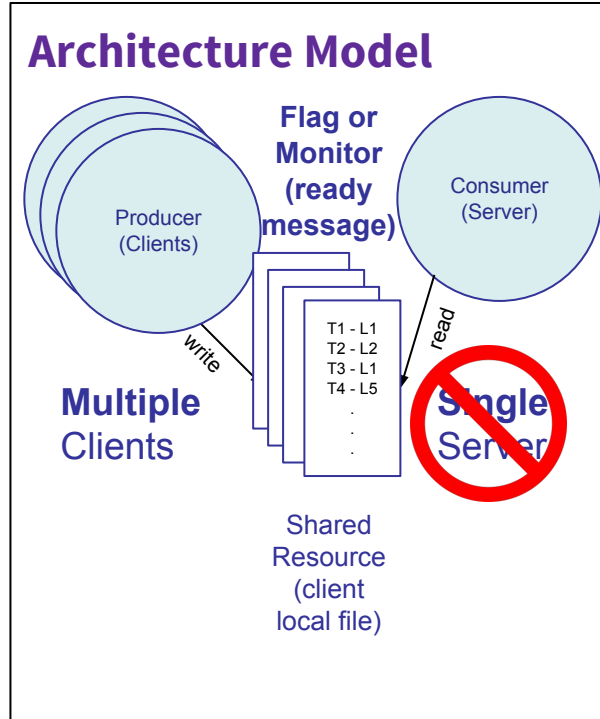
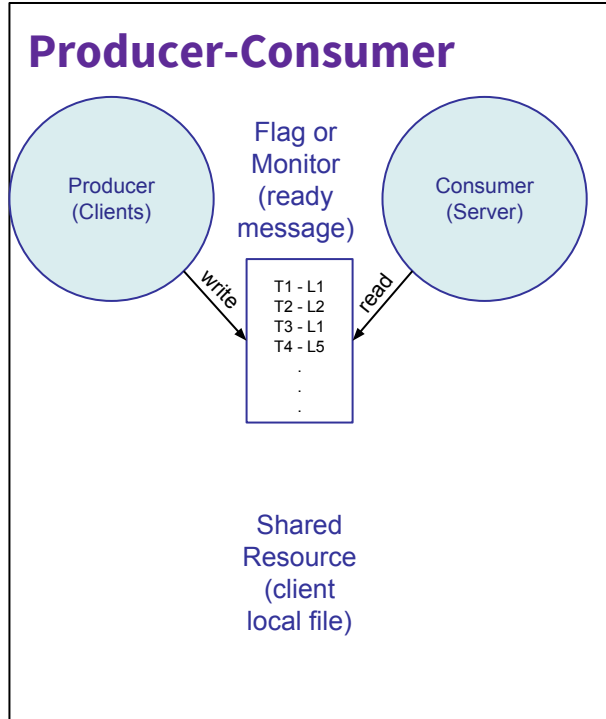


Design Approach

1. Threaded dispatch asynchronous Server
2. Use SFTP for location file transfer. TCIP communication for sending ready message

Architecture Model:

3- support server power off/message lost



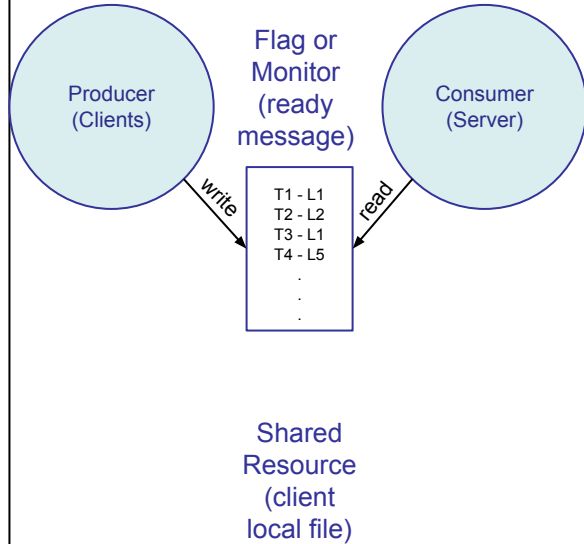
Design Approach

1. Threaded dispatch asynchronous Server
2. Use SFTP for location file transfer. TCIP communication for sending ready message
3. Update/Empty location file on 'Ready/Read' message

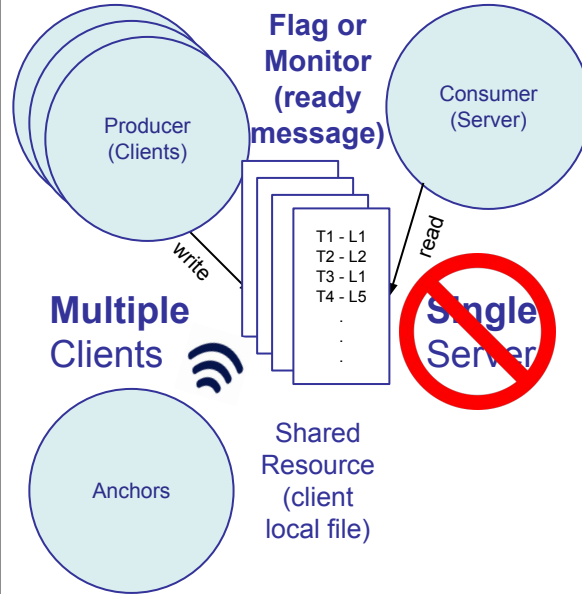
Architecture Model:

4- secure communication protocol

Producer-Consumer



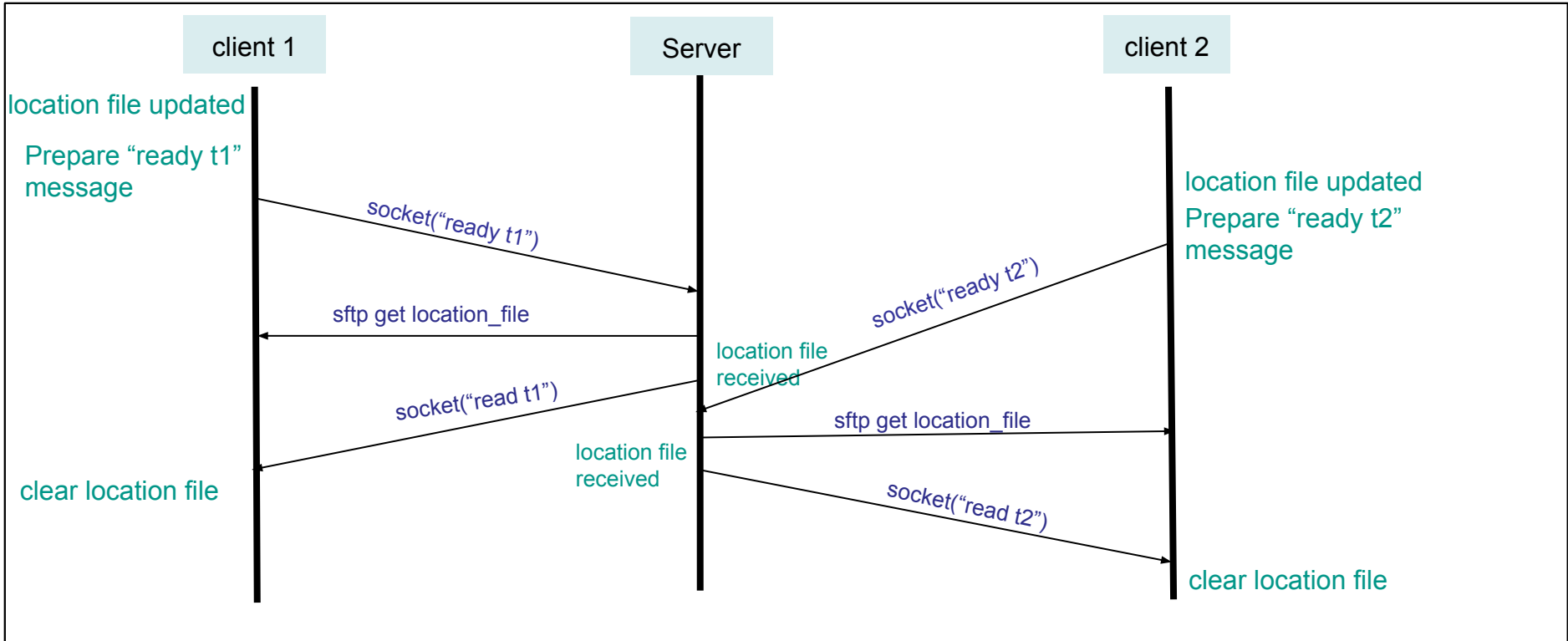
Architecture Model



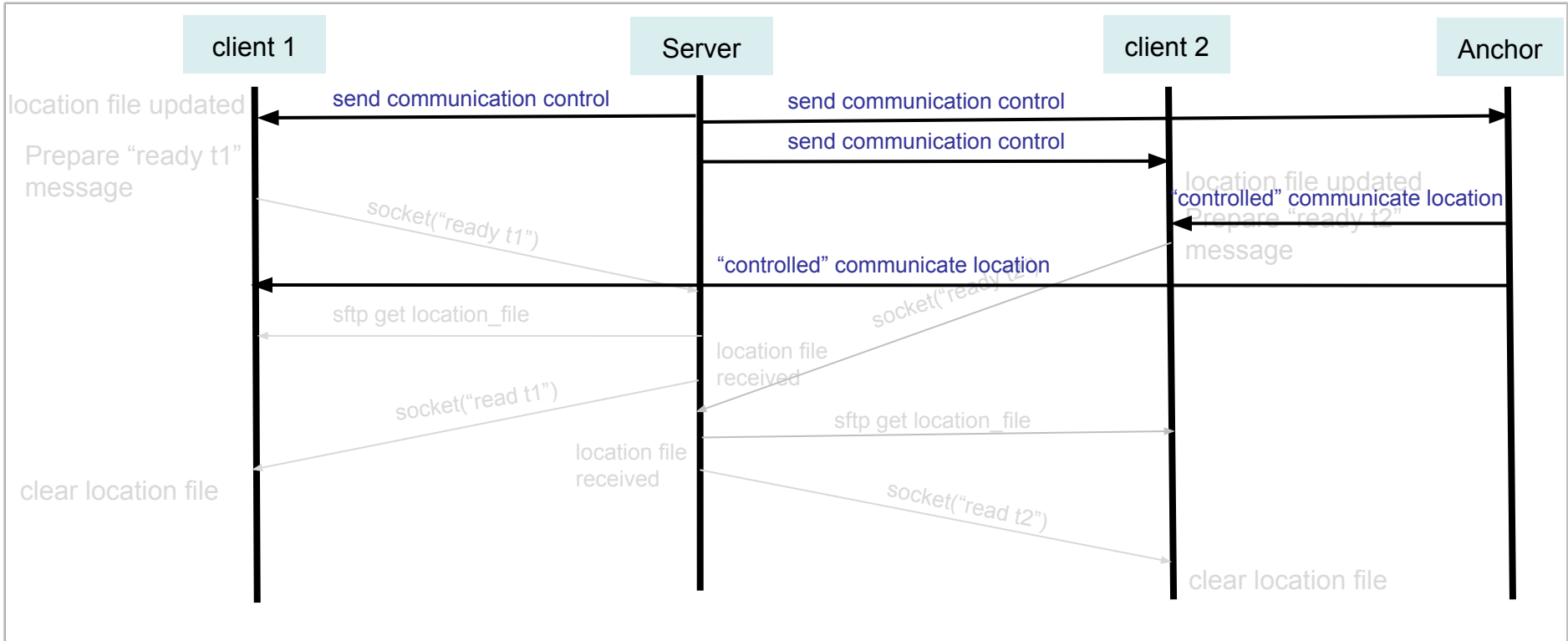
Design Approach

1. Threaded dispatch asynchronous Server
2. Use SFTP for location file transfer. TCIP communication for sending ready message
3. Update/Empty location file on 'Ready/Read' message
4. Control/Update communication protocol pattern

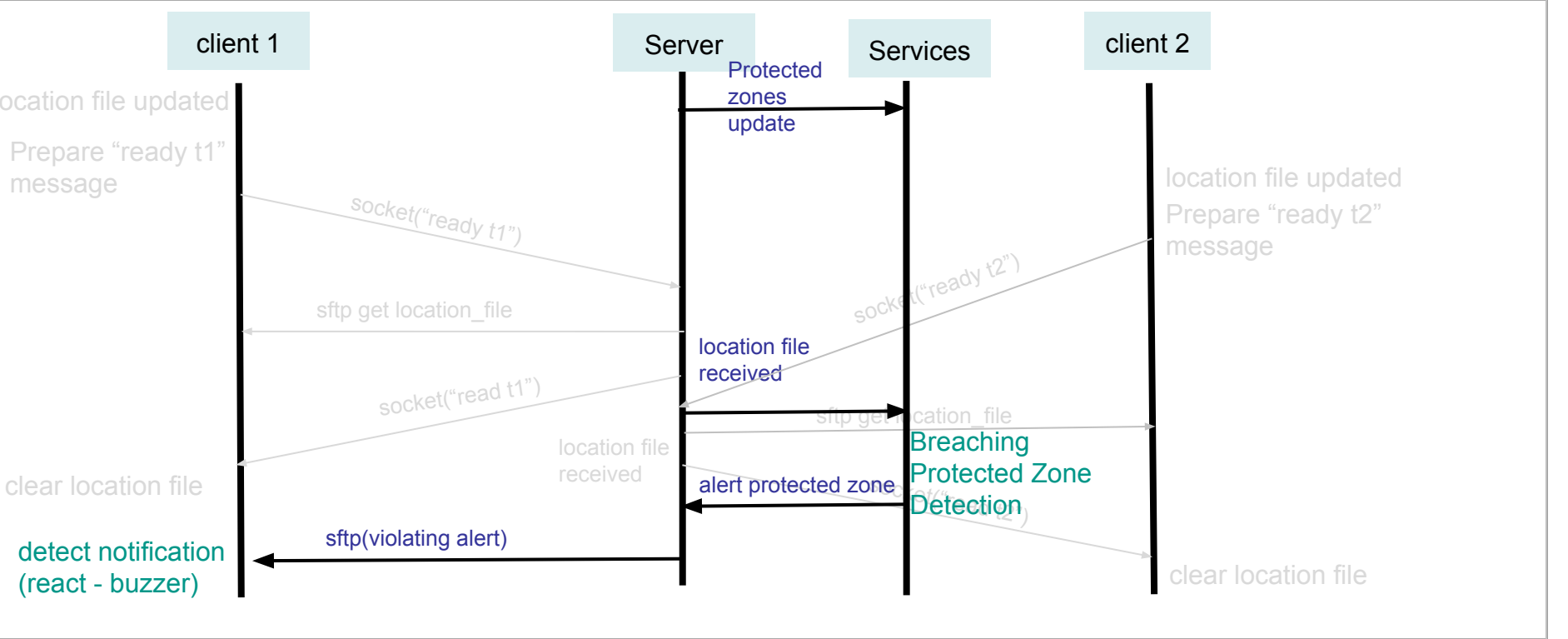
(1A) Secure Communication - "Main Protocol"



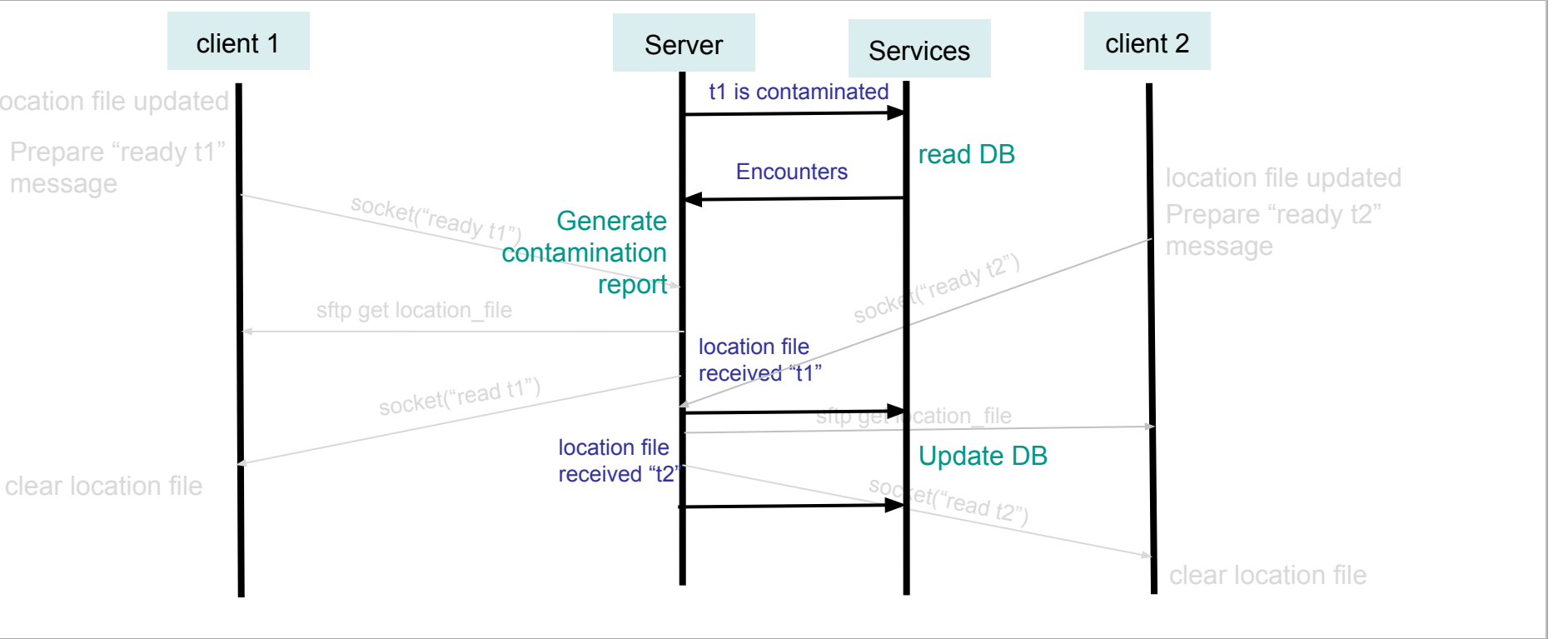
(1B) Secure Communication - “Control Protocol”



(1C) Secure Communication - “Services Protocol” - Protected Zone Alert



(1C) Secure Communication - “Services Protocol” - Contamination



(1C) Secure Communication - “Services Protocol” - Tracking

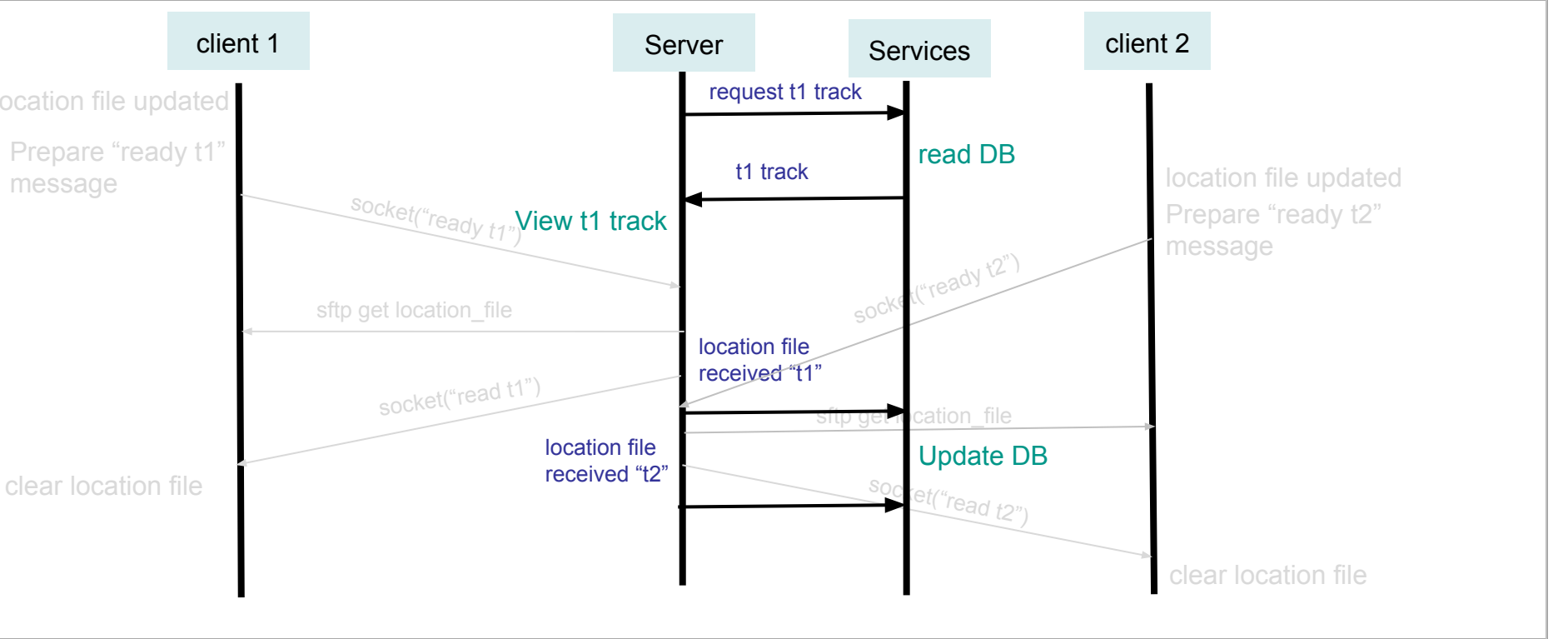


Table of Contents

1. Server Communication

2. IR Communication Security

a. Interruption

b. Interception

c. Fabrication

3. Technical Details

4. Attacks and Countermeasures

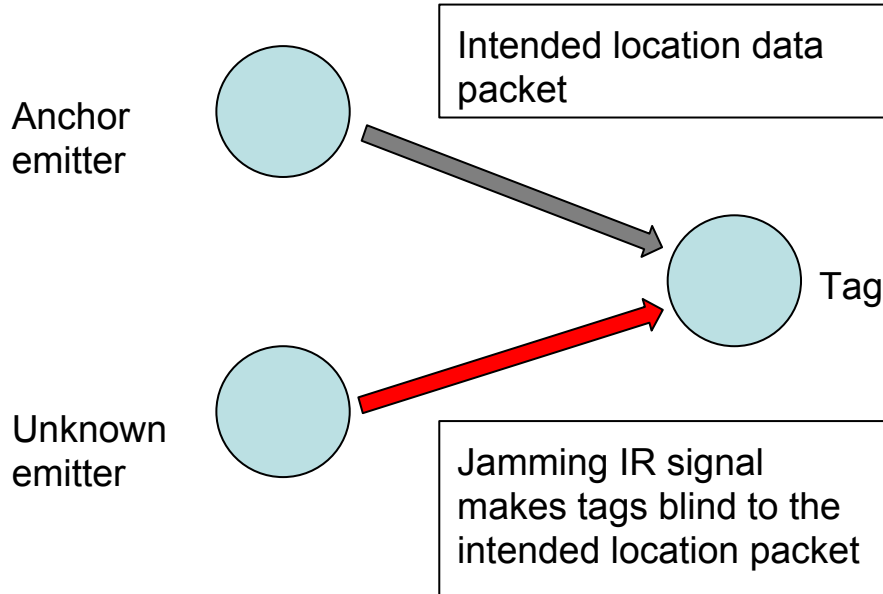
5. Future Work

6. Demo

(2A) IR Security Issues - Interruption

1. Jam IR receivers on smart tags
interrupting service to that tag, and the
security of the hospital.

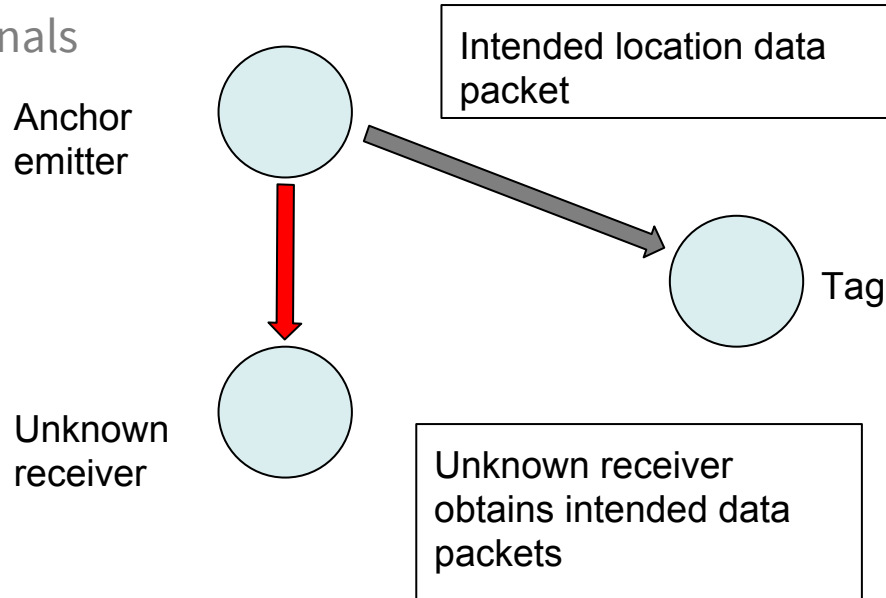
Solution: Intelligence on
part of Server /Smart Tag



(2B) IR Security Issues - The Replay Attack

2. Interception of data packets to be replayed:

- Listening to emitted IR signals



(2C) IR Security Issues - The Replay Attack

3. Replaying intercepted data packets:

- Emit a false location using previously intercepted data packets

Solution: Communication acknowledgement protocol

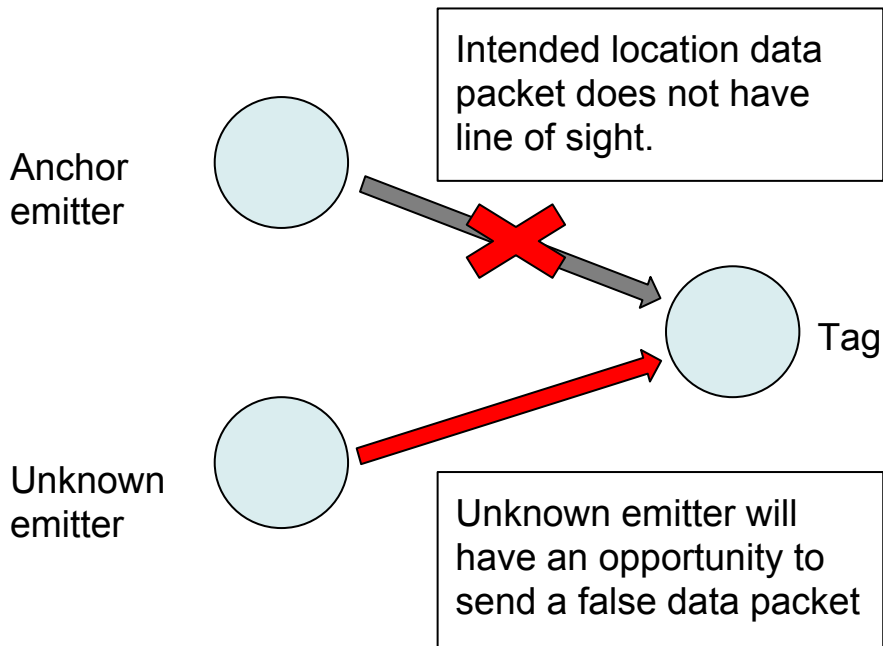


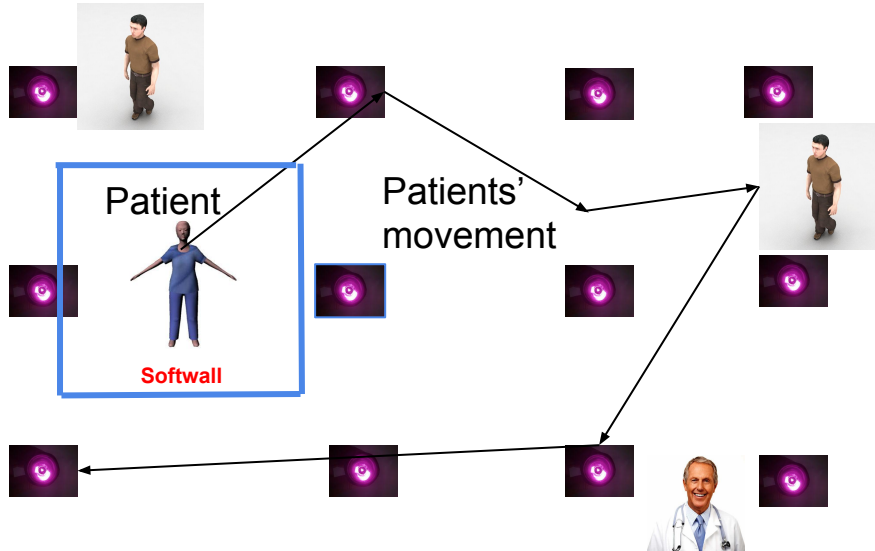
Table of Contents

1. Server Communication
2. IR Communication Security
3. Technical Details
 - a. Softwall Implementation
 - b. Improvements in IR transmission
 - c. Solving the Replay Attack
4. Attacks and Countermeasures
5. Future Work
6. Demo

(3A) Softwall Implementation

- Desirable to know whom all an infected person came in contact with
- The softwall service reads location data stored as files and determines whom all did the infected person came in contact
- Easy to figure out how infection might have propagated

Infra Red
Anchor Node



Sample Output of Softwall
service

Infected person ID = 137

Patient with tag ID 137
encountered person with
tag ID 246 at 1:45 pm ,
12th July 2015 at zone 12

.
. .
. .
. .

(3B) Improvements in IR transmission

1. Reduction in transmission duration

MID-TERM : 220 ms

NOW : 44 ms

Delays in our code were implemented with IDLE for loops -

```
for(i = LONG_DELAY; i > 0; i--);
```

PREVIOUSLY

```
#define SHORT_DELAY 350000  
#define LONG_DELAY 1400000  
#define MID_DELAY 1050000
```



NOW

```
#define SHORT_DELAY 350000/SCALING_FACTOR  
#define LONG_DELAY 1400000/SCALING_FACTOR  
#define MID_DELAY 1050000/SCALING_FACTOR
```

#define SCALING_FACTOR 5

- Lots of Experimentation done
- Scaling works fine for us !

Tried Different Scaling Factors - 2-10. Not fine for > 7

- **Smaller Transmission Duration**



More Responsive Smart Tag

(3B) Improvements in IR transmission

2. Reducing effect of outliers

Ideal preamble

20 , 20, 20 , 20 , 20 , 20 , 20 , 20 , 20 , 20, 20

Received Preamble looks like

17, 19, 22, 18, 23, 24, 22, 19, 18, 22

(each number indicates number of consecutive 1's or 0's)

- What if we receive - 17, **13** , 22, 18, 23, 24, 22, 19, 18, 22 ?



Acceptable value in a preamble is between 15 and 25

Should we therefore disregard the preamble received ?

(3B) Improvements in IR transmission

OUTLIERS NEED TO BE IGNORED !

- Receiver code modified so that it is more resilient to outliers

```
#define PREAMBLE_RELAXED_DETECTION_THRESHOLD
```

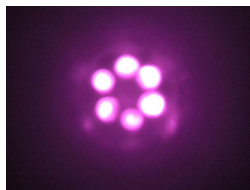
2

Number of outliers
that can be
tolerated in the
preamble

- Source of these outliers : Variability in response time of receiver, PWM pins etc ...
- But we have to live with them !
- Receiver code modified to infer preamble even presence of outliers

(3B) Improvements in IR transmission

3. Resilient Message detection



- Smart tag samples for 220 ms before it starts processes the sampled values
- Each message = 44 ms
- ~5 messages sampled in one go

EARLIER : At least 3 of the received messages need to tally

NOW : Even if one correct location is received, we communicate it to the server

Odds of randomly inferring a correct location are very low !



IR transmission is now more Resilient

(3C) Solving the Replay Attack



(I) Normally: Fixed msg from transmitter to receiver

(3C) Solving the Replay Attack



(II) Attacker Intercepts Message

(3C) Solving the Replay Attack



(III) And replays it later on!

(3C) Solving the Replay Attack

- Use an idea from cryptography - the nonce!
- Two parts
 - Random number
 - Time synchronized communication

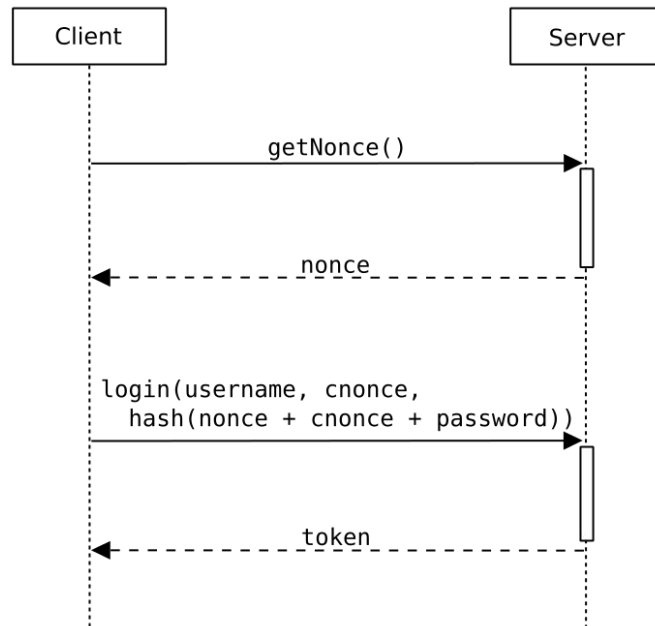
```
class AnchorsTagControl:
```

```
    start = 1  
    stop = 8  
    step = 1
```

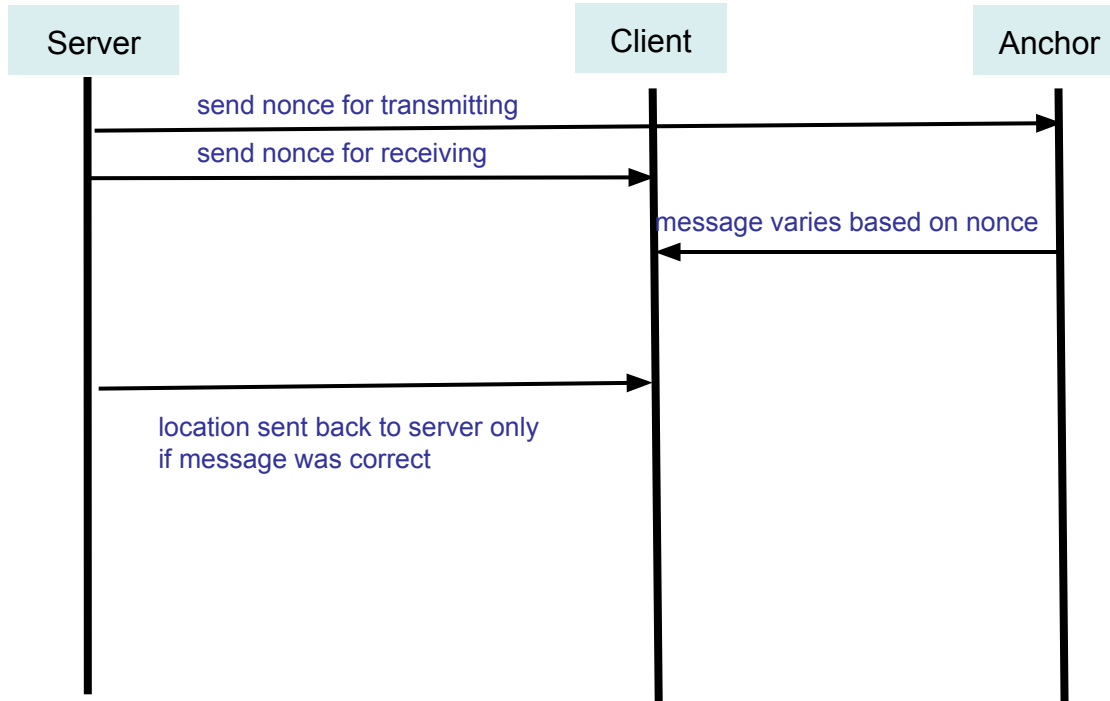
```
    def __init__(self):  
        pass
```

```
    def control(self):  
        while True:
```

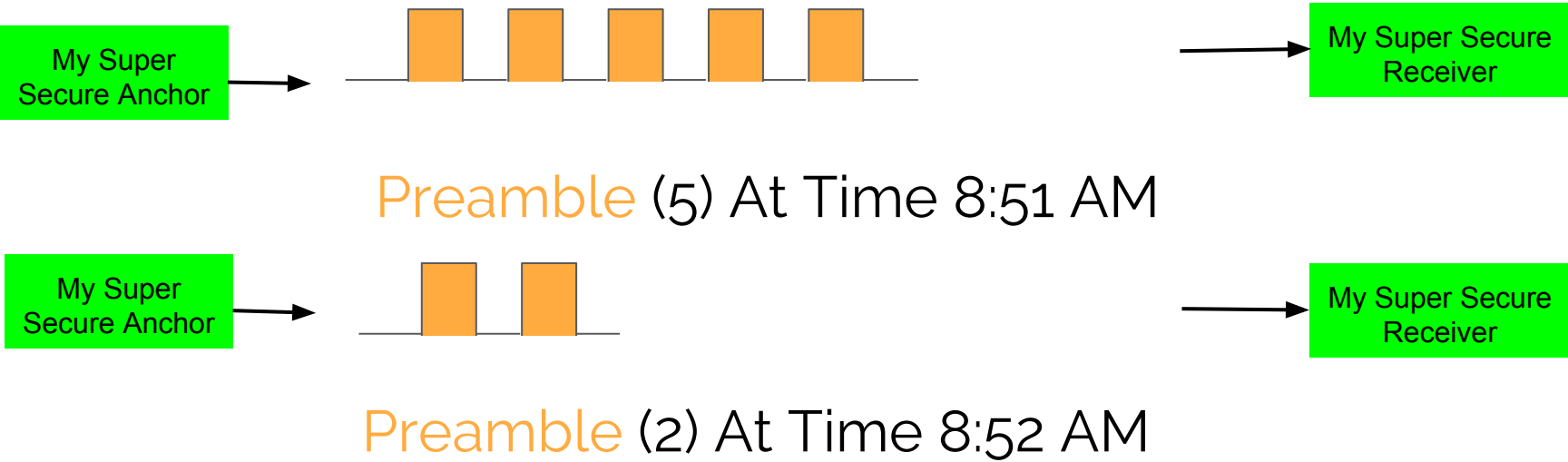
```
            # Generate random preamble length  
            random.choice(range(self.start, se
```



(3C) Solving the Replay Attack

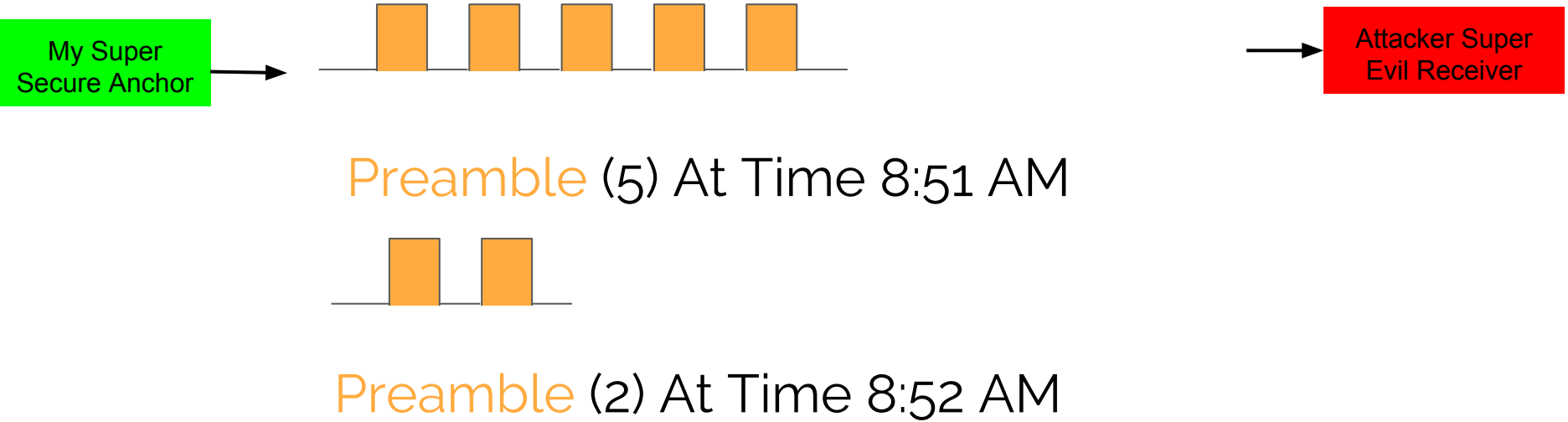


(3C) Solving the Replay Attack



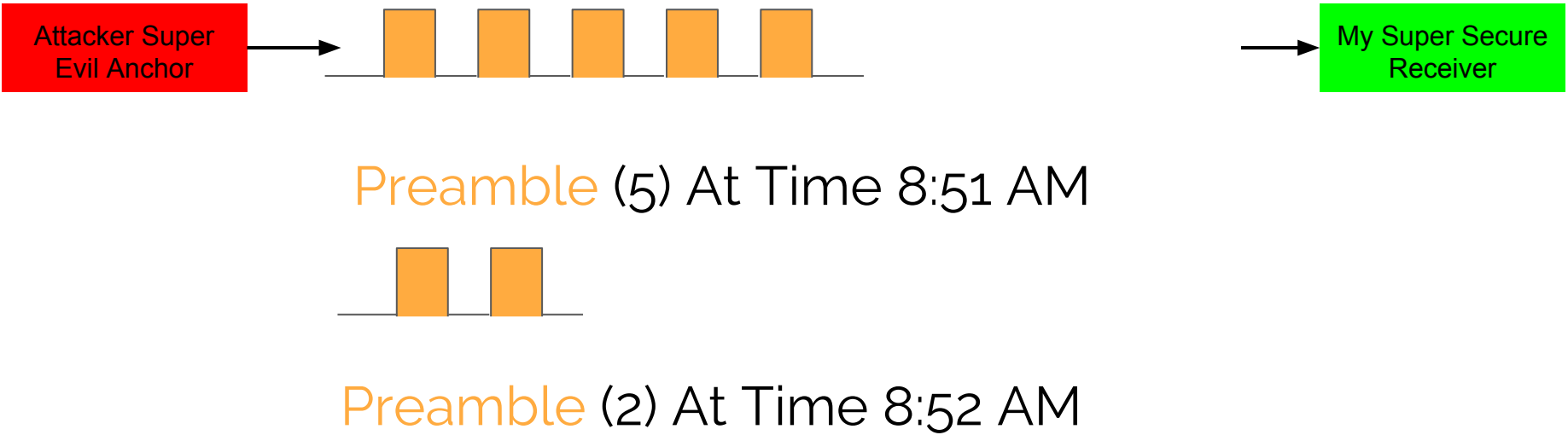
(I) Now: old messages will become invalid

(3C) Solving the Replay Attack



(II) Attacker Intercepts Message at 8:51

(3C) Solving the Replay Attack



(III) And try to replay it at 8:52

(3C) Solving the Replay Attack

Attacker Super
Evil Anchor



Preamble (5) At Time 8:51 AM



Preamble (2) At Time 8:52 AM



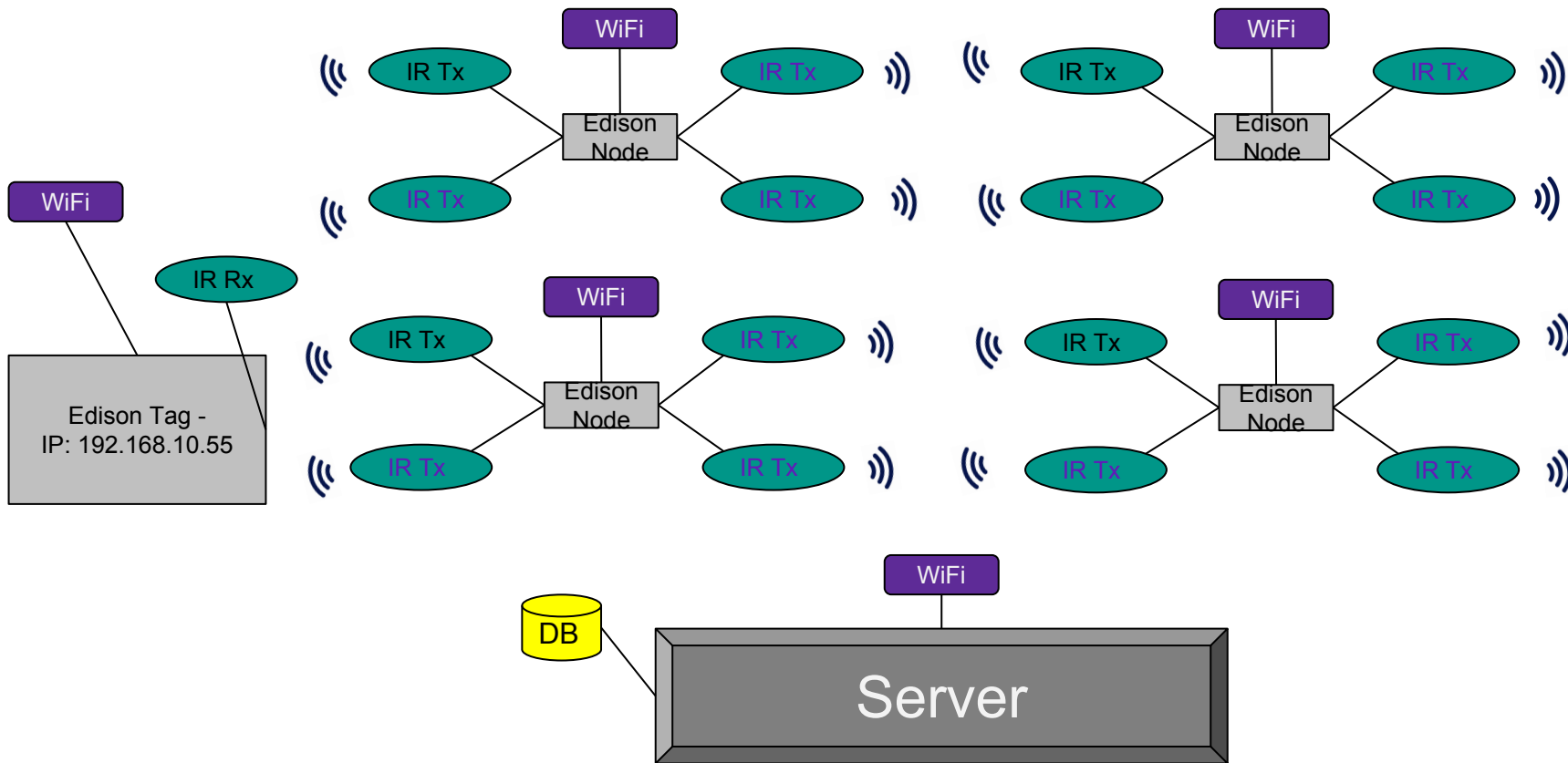
My Super Secure
Receiver

(IV) But at 8:52 that message is invalid!

Table of Contents

1. Server Communication
2. IR Communication Security
3. Technical Details
4. Attacks and Countermeasures
 - a. Privacy Invasion Attack
 - b. Jamming Attack
5. Future Work
6. Demo

(4A) Privacy Invasion Attack

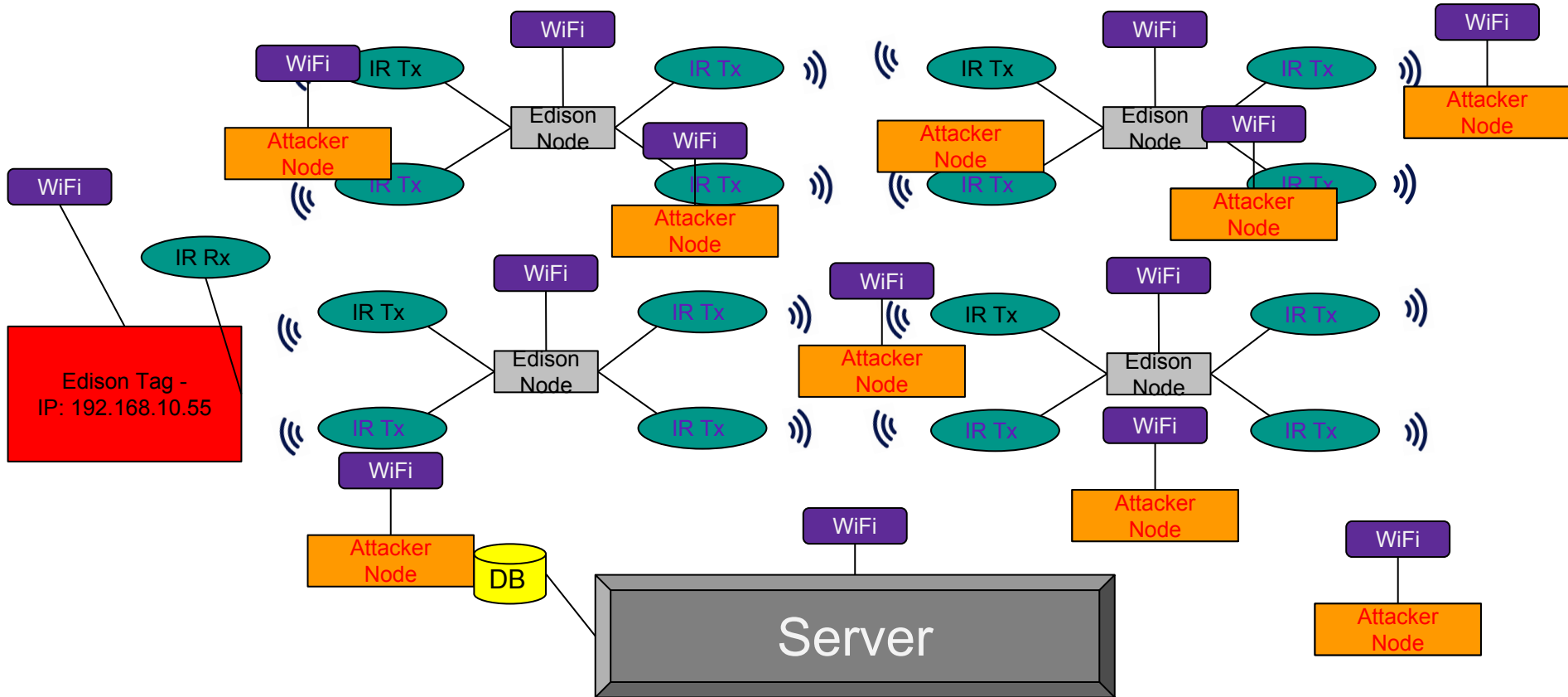


(4A) Privacy Invasion Attack

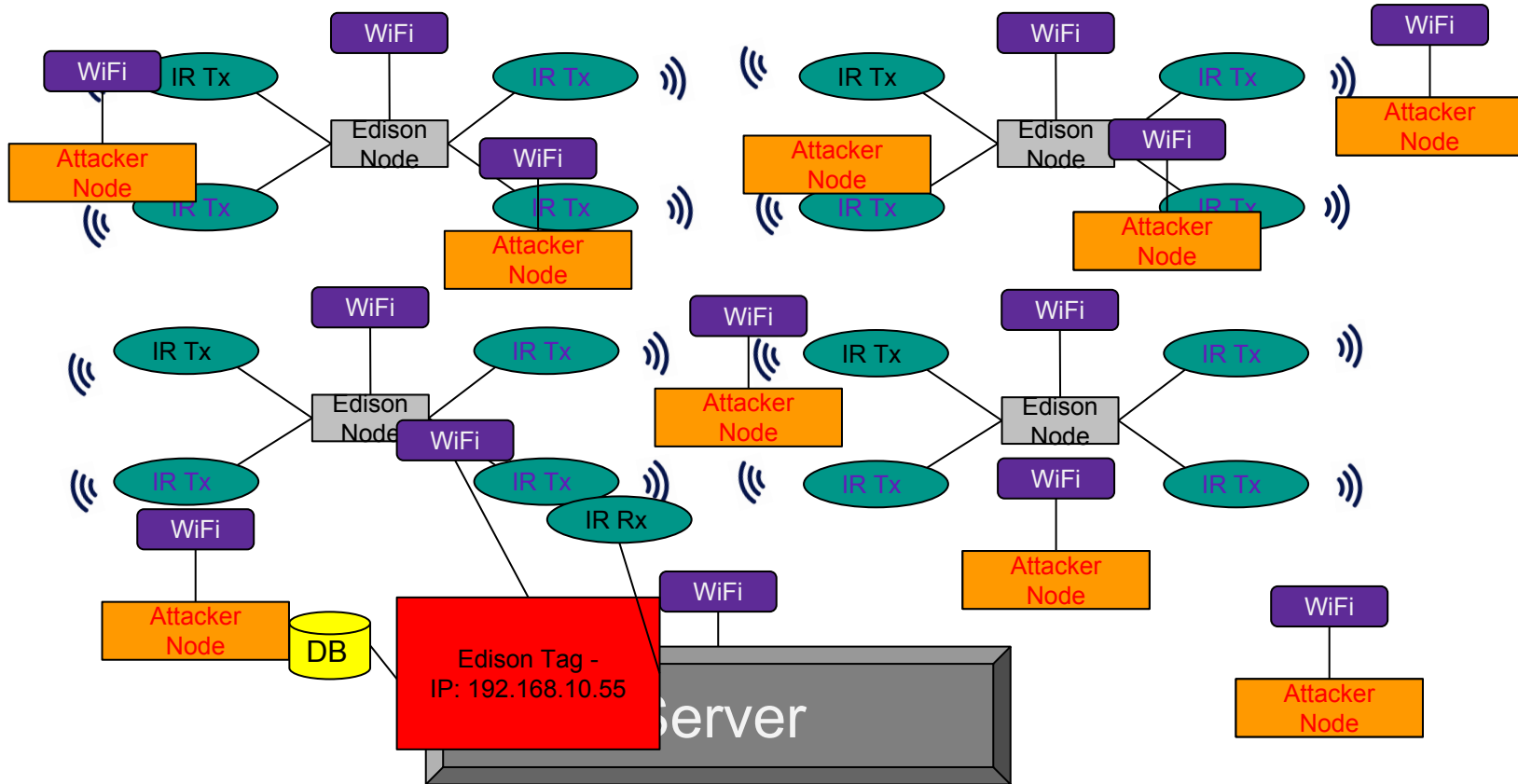
Person	IP
Joe	192.168.1.2
Sam	192.168.1.3
Amanda	192.168.1.4
Suzy	192.168.1.5
Anita	192.168.1.6

Server knows what IP is what person

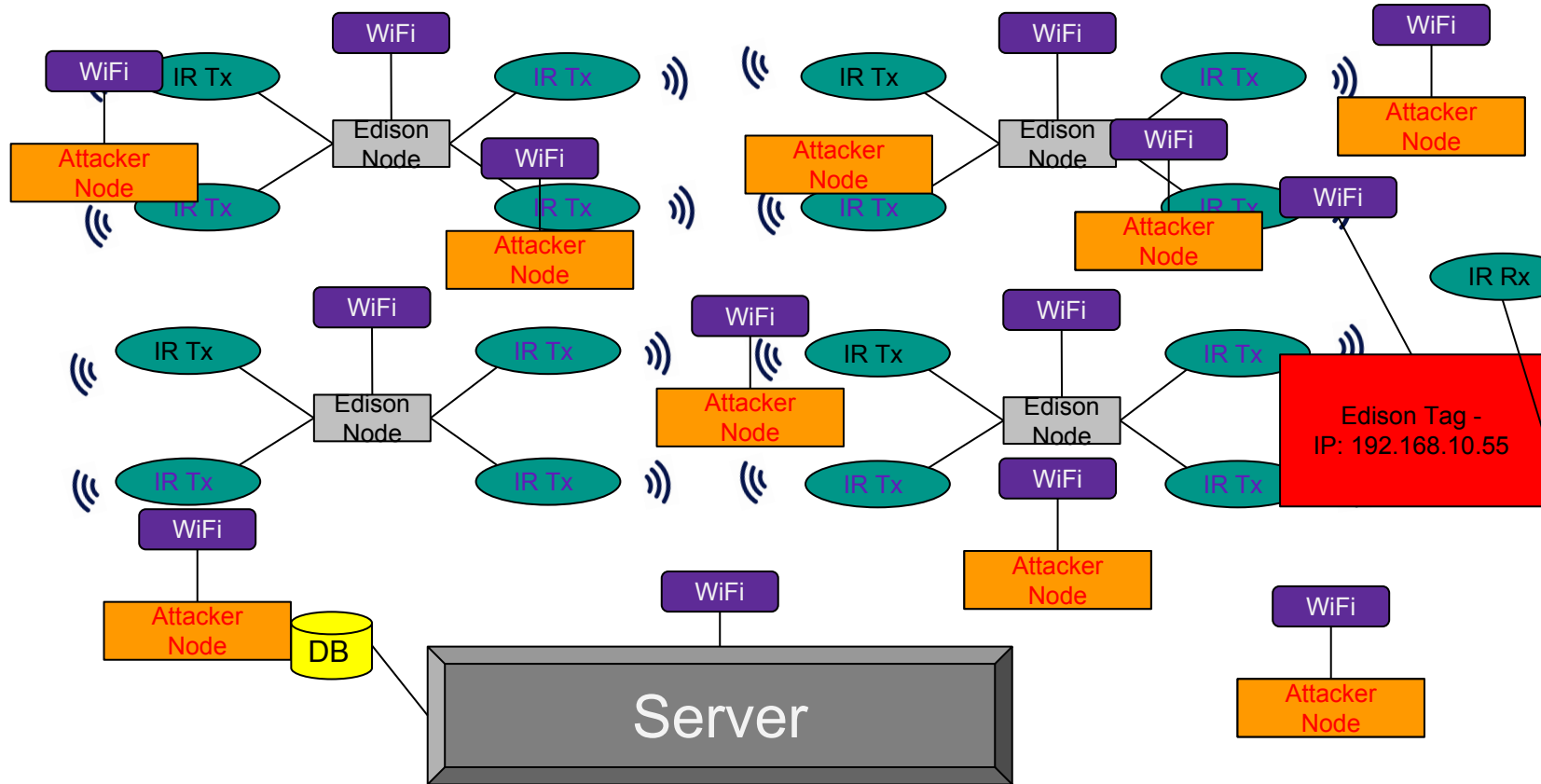
(4A) Privacy Invasion Attack



(4A) Privacy Invasion Attack



(4A) Privacy Invasion Attack



(4A) Privacy Invasion Attack



```
80/tcp    open      http
81/tcp    open      https
10.2.2.2  [mobile]
11 # nmap -u -sS -O 10.2.2.2
11 Starting nmap U. 2.5BETA25
13 Insufficient responses for TCP sequencing (3). OS detection
13 accurate
14 Interesting ports on 10.2.2.2:
44 (The 1539 ports scanned but not shown below are in state: closed)
51 Port      State      Service
51 22/tcp     open      ssh
50
60 No exact OS matches for host
60
24 Nmap run completed -- 1 IP address (1 host up) scanned
50 # ssh -o StrictHostKeyChecking=no 10.2.2.2 -rootpw="210M0101"
Connecting to 10.2.2.2:ssh ... successful.
Re Attempting to exploit SSHv1 CRC32 ... successful.
IP Resetting root password to "210M0101":
System open: Access Level (9)
50 # ssh 10.2.2.2 -l root
root@10.2.2.2's password: [REDACTED]
```

RTT CONTROL
ACCESS GRANTED

(4A) Privacy Invasion Attack

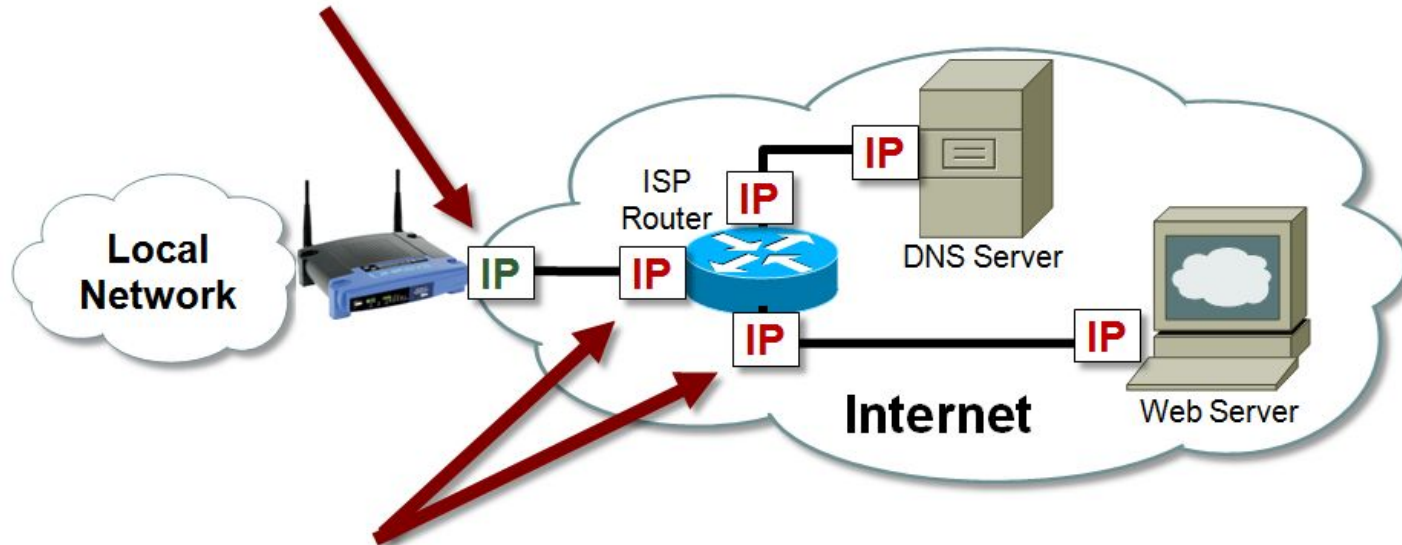
Attack Node	IP
5	192.168.1.2
7	192.168.1.3
8	192.168.1.4
9	192.168.1.5
1	192.168.1.6

(4A) Privacy Invasion Attack

Attack Node	IP	Observed
5	192.168.1.2	Joe
7	192.168.1.3	Sam
8	192.168.1.4	Amanda
9	192.168.1.5	Suzy
1	192.168.1.6	Anita

(4A) Privacy Invasion Attack

- Dynamic IP addresses periodically change
 - Typically assigned to ISP customers



- Static IP addresses never change

(4A) Privacy Invasion Attack

Attack Node	IP	Observed
5	192.168.1.2	Joe, Sam
7	192.168.1.3	Sam, Joe
8	192.168.1.4	Amanda, Anita, Suzy
9	192.168.1.5	Suzy, Anita, Amanda
1	192.168.1.6	Anita, Sam, Joe

(4A) Privacy Invasion Attack



4. **\$ vi sms.py**
5. Type the following code.

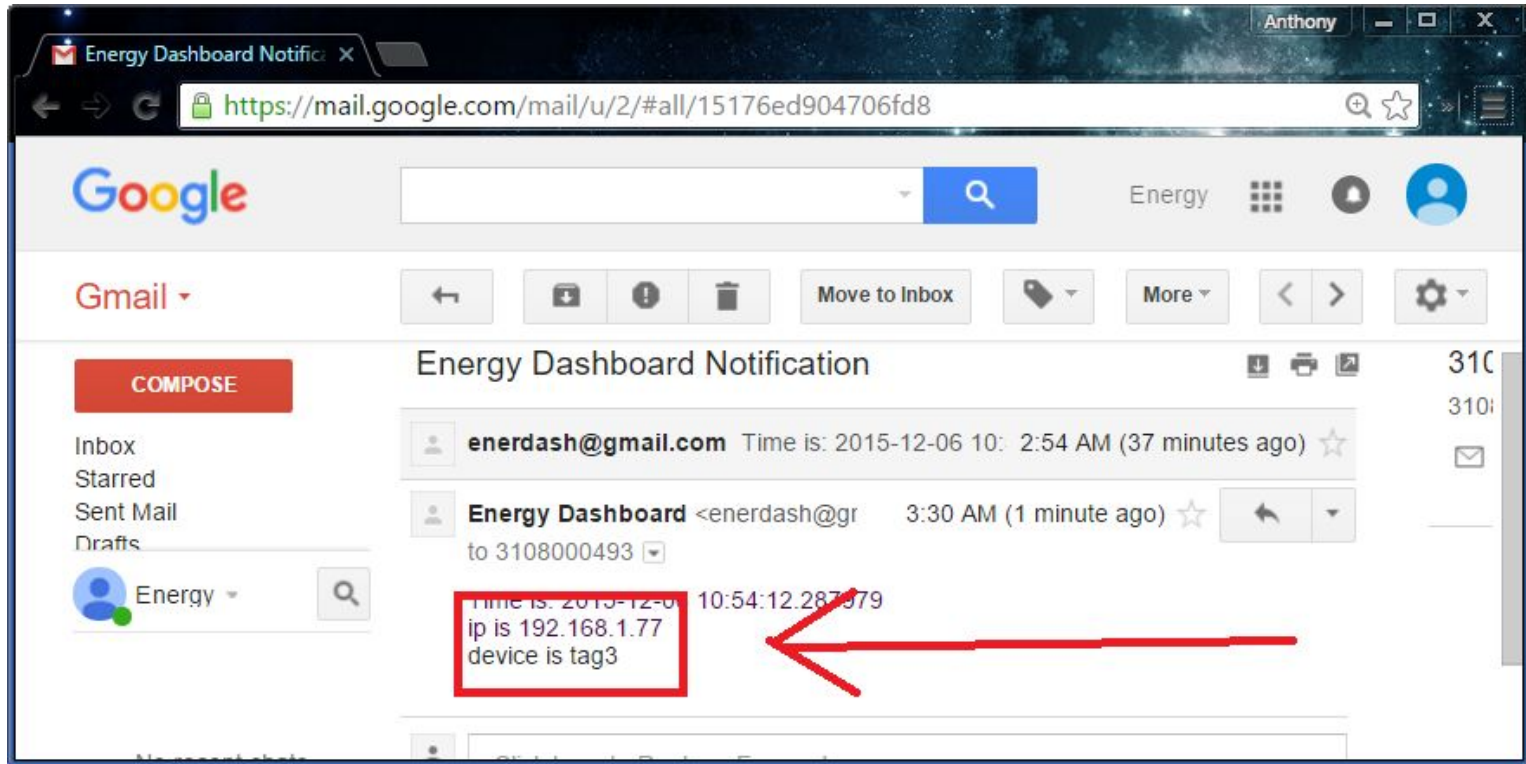
```
#!/usr/bin/python
import smtplib

server = smtplib.SMTP('smtp.gmail.com', 587)
server.ehlo()
server.starttls()
server.ehlo()
server.login("your_email_address@gmail.com", "your_password")
msg = "\nText sent from my Edison!!"
server.sendmail("your_email_address@gmail.com", "your_phone_number@vtext.com", msg)
```

Figure 8 SMS message example

Tag should inform server of IP change...

(4A) Privacy Invasion Attack



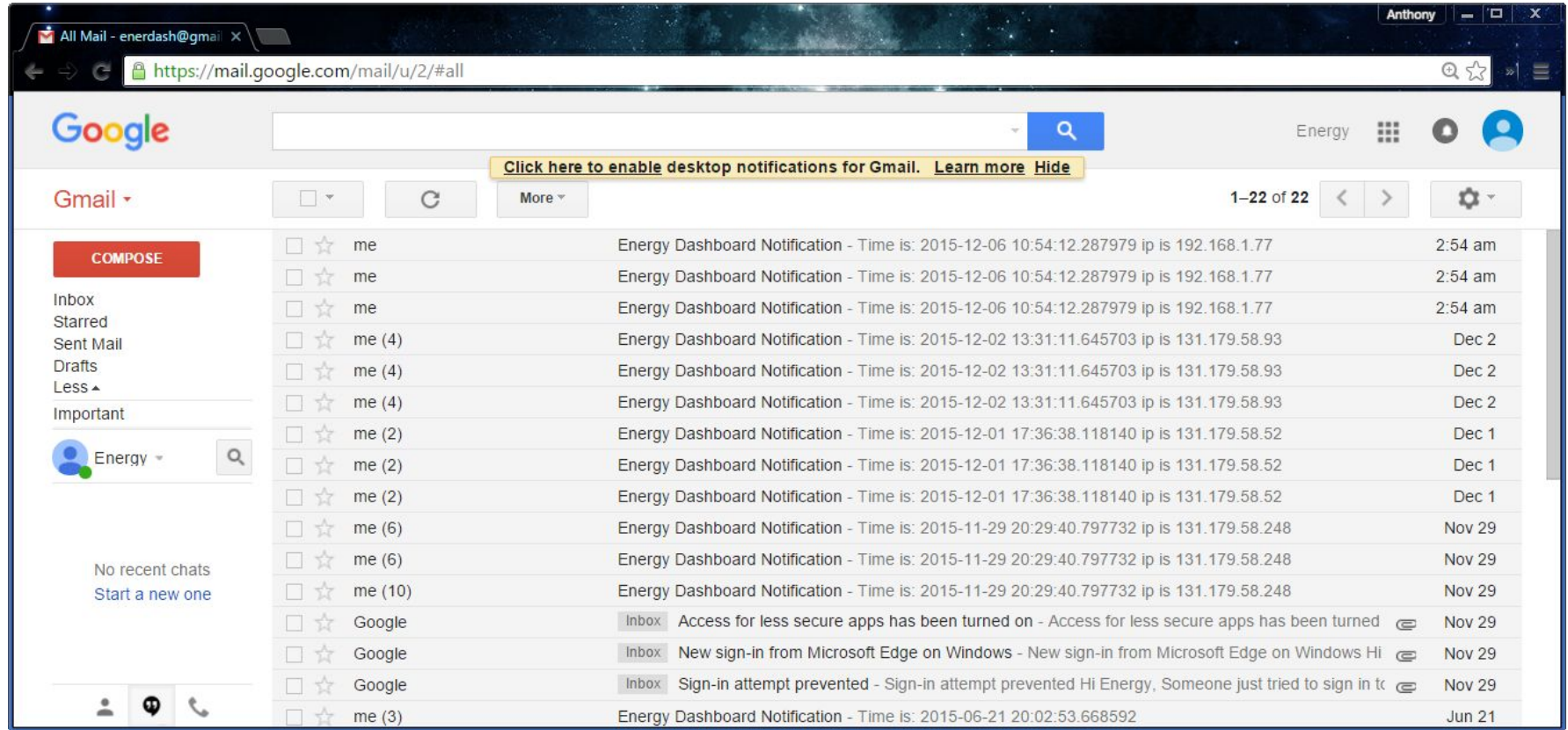
message should tell both the IP and Identity

(4A) Privacy Invasion Attack

```
Starting Cleanjournal service...  
[ OK ] Started Cleanjournal service.  
Starting report_ip.service...  
[ OK ] Started Network Service.  
[ OK ] Started File System Check on /dev/disk/by-partlabel/home.  
[ OK ] Started Permit User Sessions.  
[ OK ] Started Login Service.
```

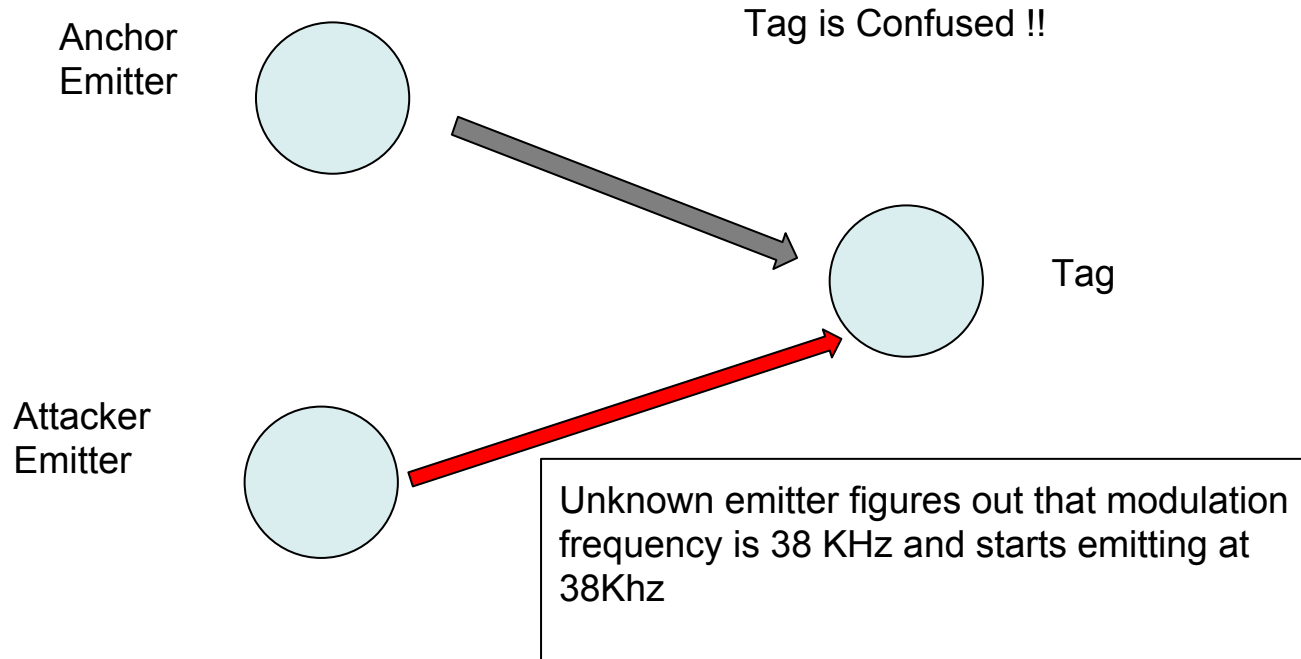
and do it every time it runs the Linux boot sequence!

(4A) Privacy Invasion Attack



and it's available even if server is down

(4B) Jamming Attack



(4B) Jamming Attack

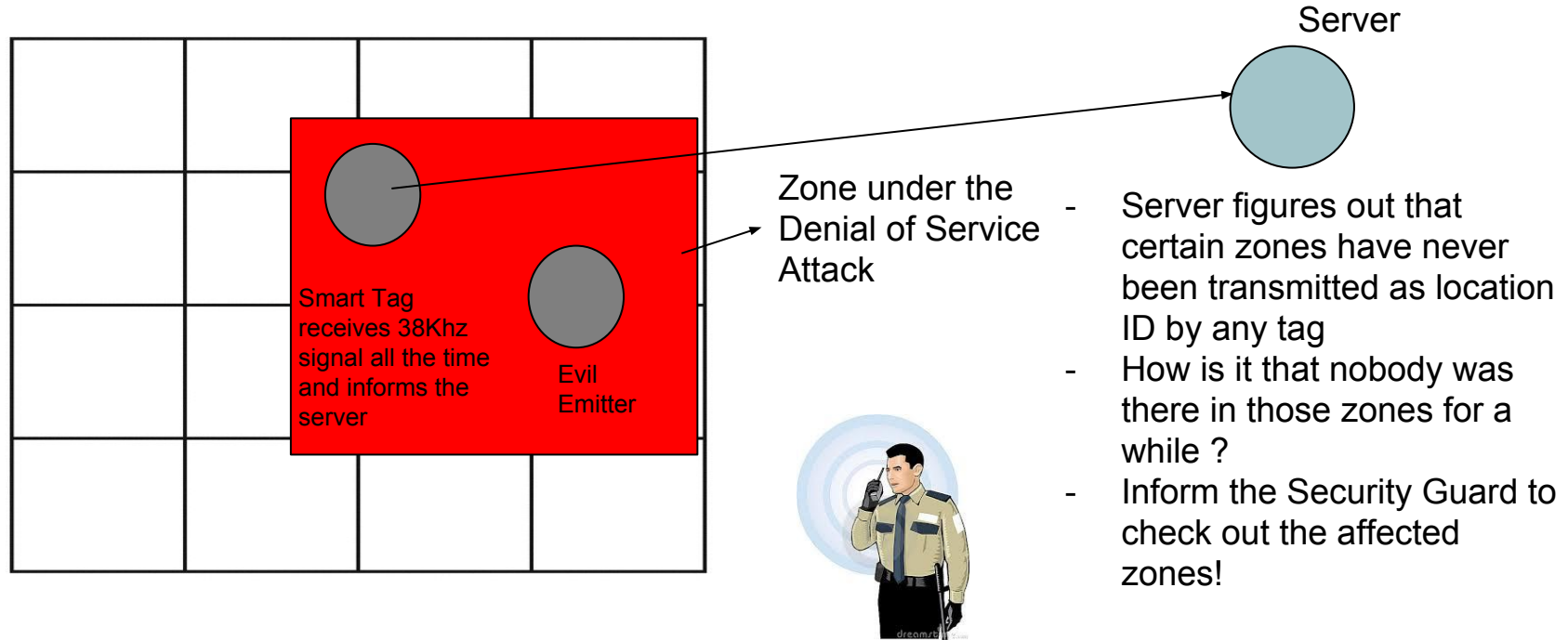


Table of Contents

1. Server Communication

2. IR Communication Security

3. Technical Details

4. Attacks and Countermeasures

5. Future Work

- a. Improving Receiver Line of Sight
- b. Enhancing Responsiveness
- c. Enhancing Cost Efficiency
- d. Practicality Aspects of the SHCP

6. Demo

(5a) Improving Line of Sight Problem

Infra-red communication suffers from line of sight problem

- Can we have **arm-bands** instead of smart tags being worn around the neck?
- Can we bundle IR leds together to increase the angle of the Infrared cone



(5b & 5c) Enhancing Responsiveness and Cost Efficiency

(5B) Reduce Duration of Transmission

- Use MCUs to reduce transmission time further down to < 1 ms



INCREASED RESPONSIVENESS OF THE TAGS

(5C) Reduction in number of Intel Edison

- Use GPIO pins of the Edison to emulate the PWM pin
- Delays can be implemented using IDLE for loop delays



INCREASE IN COST EFFICIENCY

(5d) Practicality aspects of SHCP

Range of IR emitters

- For the demo, we have implemented low range IR leds
- In actual , if the leds are mount on the ceilings
- The IR beam needs to be powerful enough to reach the smart tags
- We need to have **higher power IR leds**
- The beam of higher power IR LEDs is still focussed and hence beams from adjacent IR LEDs will not overlap



High Power IR LED Beam (seen through night vision camera)

(5d) Practicality aspects of SHCP

Power Supply to the LEDs

- Use of long wires attached to the ceiling
- One time installation matter !

Repair/Maintenance issues

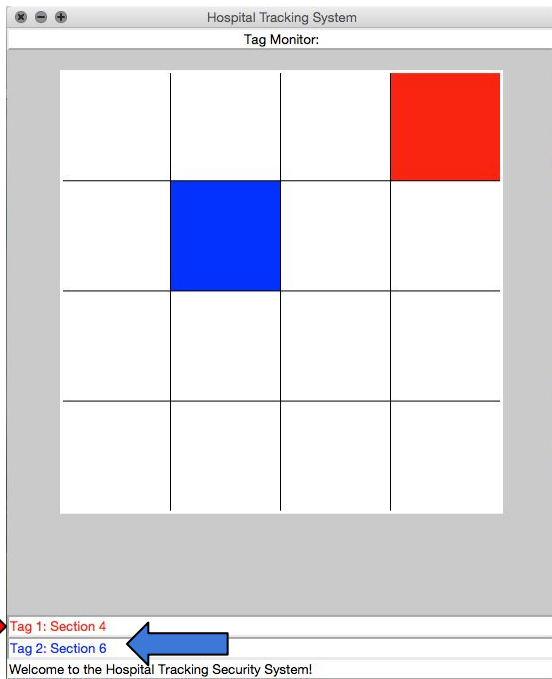
- Server can intelligently figure out the failed IR LEDs
- Server can run a **test program** during maintenance hours
- Known tag moved around the hospital with a service person
- Robust design of Infrared LEDs

Smart tag concealed by a person

- What if a person conceals a smart tag ?
- The server can detect if a smart tag goes out offline
- Send an alert message to the tag
- A buzzer or a pre-recorded voice message

Table of Contents

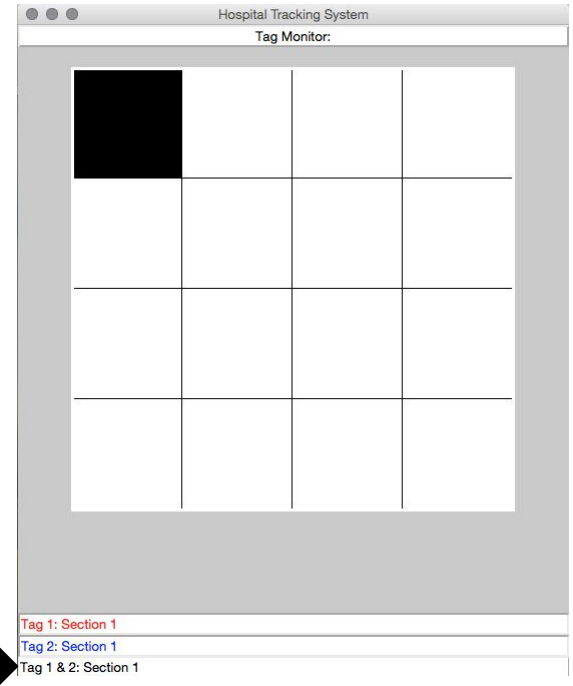
1. **Server Communication**
2. **IR Communication Security**
3. **Technical Details**
4. **Attacks and Countermeasures**
5. **Future Work**
6. **Demo**

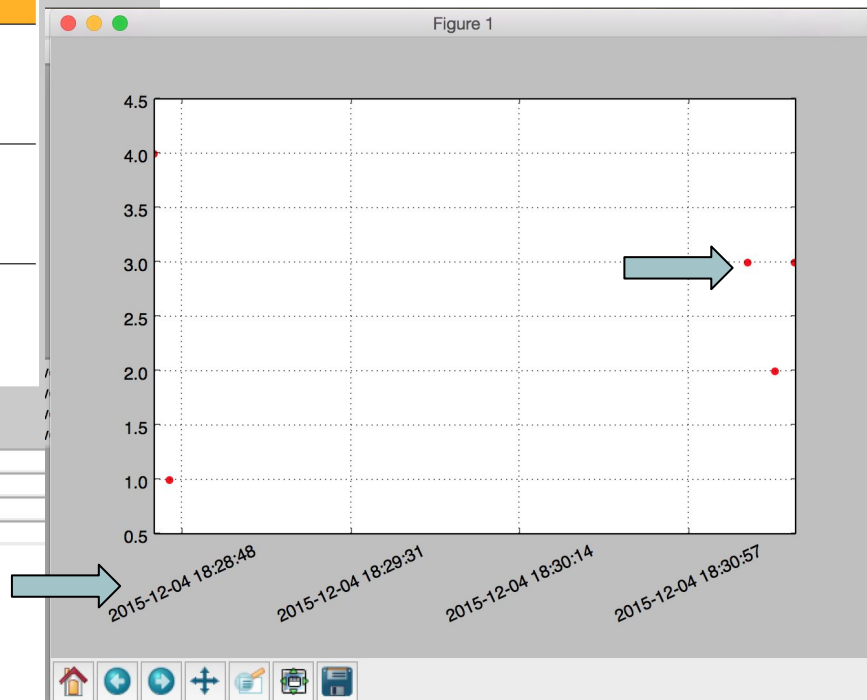
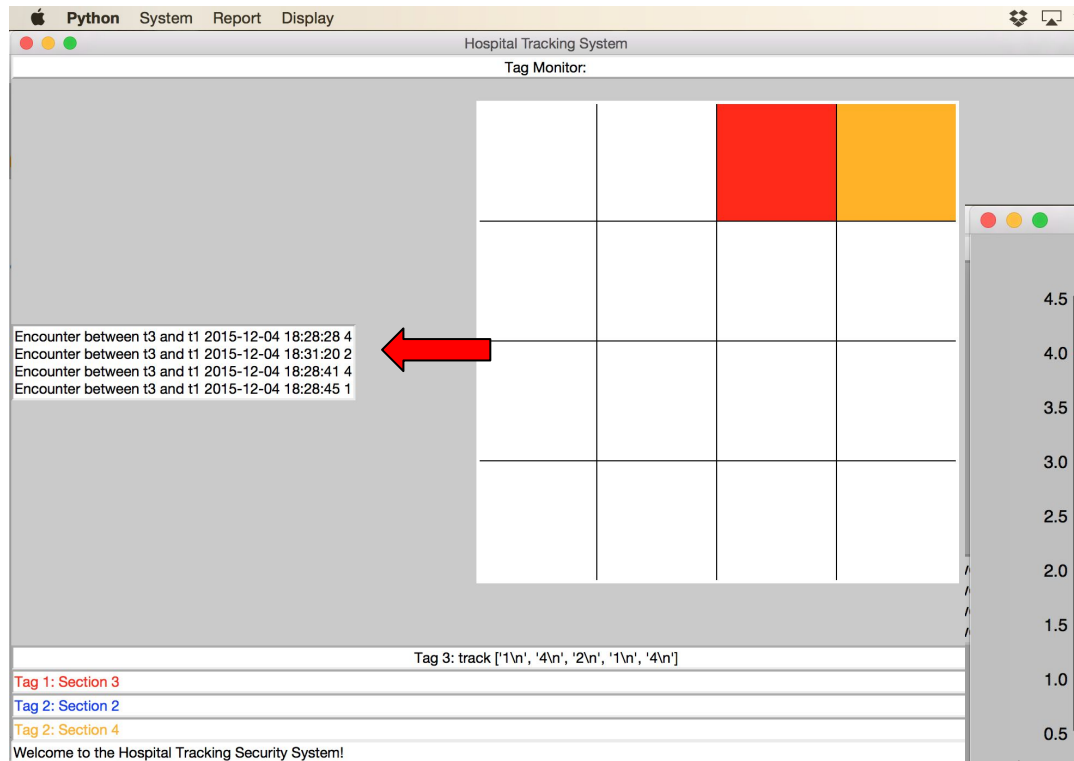


The GUI displays locations in real-time.
It is color coded for user friendliness.

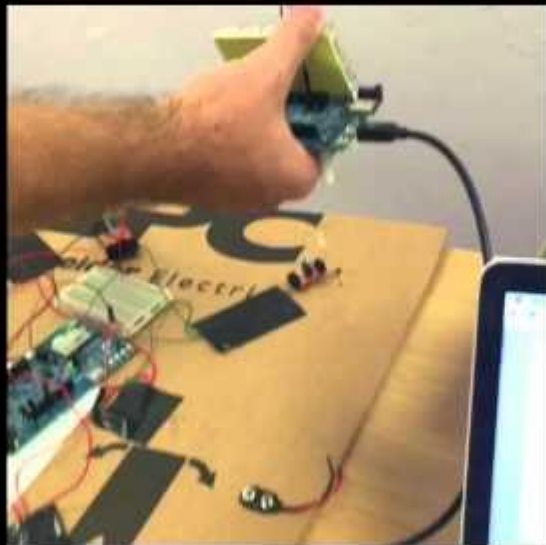
Real-time tag collision is detected and shown by a section containing more than one tag turning black.

The status bar at the bottom of the GUI will show which tags exist in the black section.





Demo



Q & A

Thank You!

Team Responsibilities - Salma

1. Set the threat model and the system guarantees of the system.
2. Designed and implemented the back-end asynchronous dispatch server along with the client code to handle the communication protocol in a secure way.
3. Designed a communication protocol to maintain the integrity of the information by acknowledgment scheme between the server and the clients.
4. Designed and implemented the server services including:
 - a. Tracking map for all the tags
 - b. Protected zone handling
 - c. Encounters history for contamination
5. Designed and implemented three layers of communication between the server and the clients:
 - a. Main protocol to communicate the location between the clients and the server
 - b. Control protocol for updating/controlling the communication scheme between anchors and tags
 - c. Services protocol to handle the server services and communicating the actions needed to the tags (ex. alerting a violating tag in a certain location)
6. Worked on integrating the server code and the client code with the IR receiver code.
7. Worked on integrating the server code with the GUI.
8. Added support of the services in the GUI.
9. Maintained a special data structure on the server that can hold the time series of the locations for each individual tag in a way that makes the query fast. This database is updated at runtime in parallel thread in order not to harm the responsive of the server.

Team Responsibilities - Raymond

1. Created GUI and was responsible for debugging and scaling of the GUI.
 - a. Created the algorithms necessary for multi-threaded tracking of tags.
 - b. Designed and refined the algorithms used to display tags on the GUI.
 - c. Created the menu bars which allowed for simple integration of features into the GUI.
2. Wrote the original IR emitter/receiver algorithms.
 - a. Collaborated with group members to expand those algorithms to what was implemented during the final project.
3. Designed the MOSFET driver circuits and was responsible for debugging issues involving either the emitter or receiver circuitry.
 - a. Experimented with different driving currents to establish the appropriate design for the project.
 - b. Discovered and resolved hardware issues that were causing the IR communication to become unreliable.
 - c. Wrote a C code testbed for the MOSFET drivers, emitters, and receivers.
 - i. This allowed for the observation and confirmation, via Oscilloscope, of two critical system aspects.
 - The first being the inability of the IR receivers to operate correctly when too close to the emitters (this distance depends on emitter output power).
 - The second was the interference between two transmitted data packets corrupting the received signal. Making understood the importance of the design tradeoff between the emitter output power, and the spacing between the emitters.
4. Worked on the integration of all aspects of the project.
 - a. Collaborated with group members to get the server to receive a tag's location in real-time.
 - b. Collaborated with group members to get the GUI to display the values received by the server in real-time.

Team Responsibilities - Anthony

1. Exploration of database services appropriate to store and maintain the private location data that would be collected by
 - a. This contribution was not relevant to our demo because we did not develop any features that require tracking over a long period of time. This could be discussed further under Future Work.
2. Extensive collaboration with other group members developing the IR emitter/receiver algorithms.
 - a. Collaborated with group members to expand those algorithms to what was implemented during the final project.
 - b. Individually added components to allow the system to dynamically change the IR message transmitted between Emitter and Receiver, preventing replay attacks.
3. Development of System Scalability optimizations
 - a. Developed an automated method to add new Tags and Anchor Nodes into our system so that the server is aware of them.
 - i. Key to this was learning how to configure programs on the Edison to run upon boot time to allow for non-interactive configuration
4. SparkFun Development
 - a. Performed investigations into using the GPIO pins on the SparkFun breakout boards. Learned the Linux convention to set up, enable and disable the pull-up and pull-down resistors on the Edison GPIO pins. Configure Multiplexers on the SparkFun blocks to enable PWM output.
5. MCU Development
 - a. Developed code based on tutorials provided by Chris that would allow for greatly shortened IR message length in the hundred-microsecond range
 - b. Code was developed as a side feature but we did not complete integration with the main system.
6. Worked on the integration of all aspects of the project.
 - a. Collaborated with group members assemble the hardware portion of the demo so that we could transmit the values to GUI on the server in real-time.

Team Responsibilities - Pranjal

1. Designed and developed the Infrared Communications Protocol
 - Initially proposed Infrared Communication as the suitable localization technology to achieve our project mission
 - Came up with the appropriate preamble and bits patterns for our protocol , in consultation with other team members
 - Designed and Implemented the Infrared Receiver Algorithm which is being used for project currently
 - Helped in debugging bugs in the transmit code
2. Worked on enhancing the responsiveness and robustness of the smart tags
 - Designed and implemented scheme for improving robustness of the IR communication towards outliers .
 - Designed and implemented scheme for having a more resilient smart tag and overcome the line of sight problem
 - Implemented the scaling factor for bringing down the transmission duration, thereby making the smart tag more responsive
 - Rigorously tested the IR communications to ensure robust operation
3. Worked on hardware assembly for the final demo
 - Helped in setting up the SparkFun boards for driving the infrared Leds
 - Did experimentation in collaboration with Raymond to figure out the correct spacing between the Leds for the purpose of our demo
 - Debugged issues related to assembly and system integration
4. Collaborated with team members in integration of GUI and the smart tag
5. Developed security scheme for averting jamming attack on the smart tags
6. Collaborated with team members to identify security threats in Infrared Communications and helped in testing the security aspects
7. Designed and developed software for implementing softwall service for our project .This service enables the doctor to track spread of contamination from an infected person to others. This service is the key utility to aid the doctors in averting hospital acquired infections

Appendix

Needs for a Scalable System

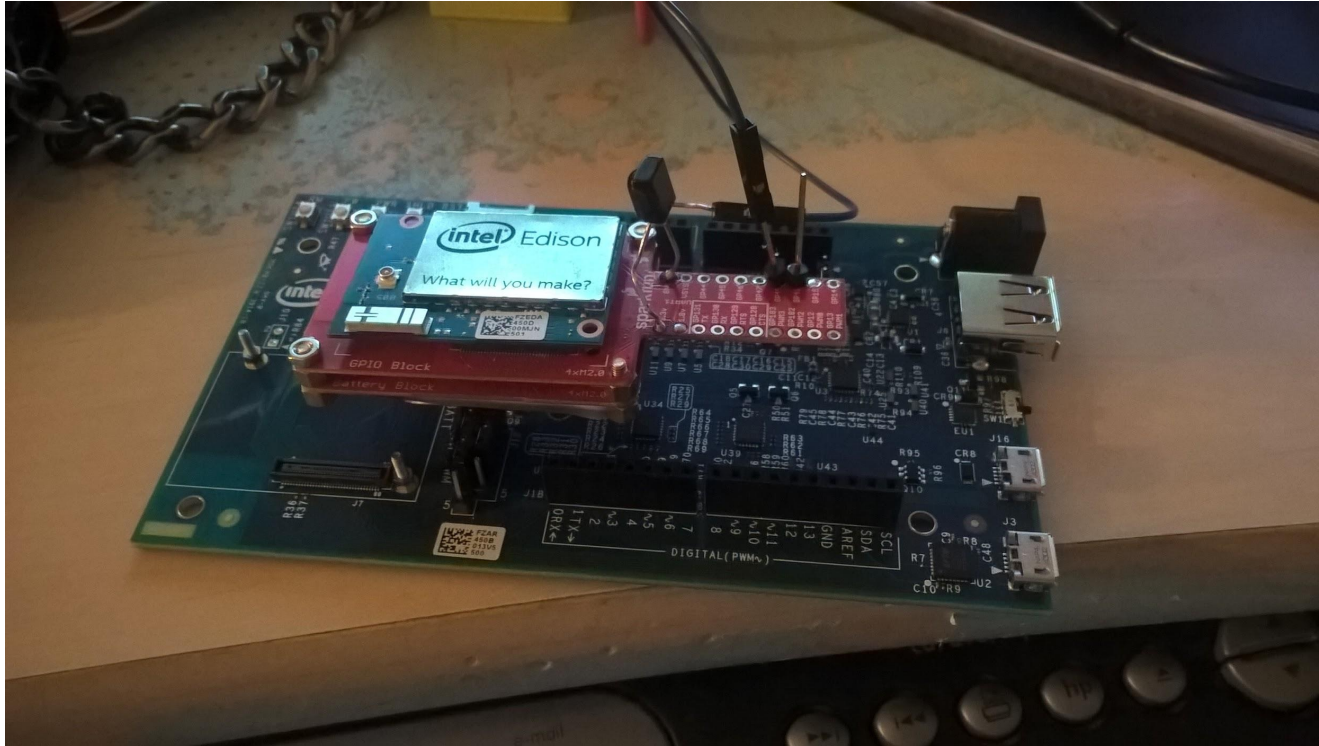
- Anchors
 - Constructs a new building/wing
 - Replace faulty equipment
- Tags
 - Hires new employees
 - Admits new patients
 - Allows new visitors
- Server
 - New people (security guards, management) allowed access to view tracking info
 - New server side services (reports/visualization/zones)

Automated Installation for Anchors and Tags

Key Information that the Server requires for all new Edison system nodes

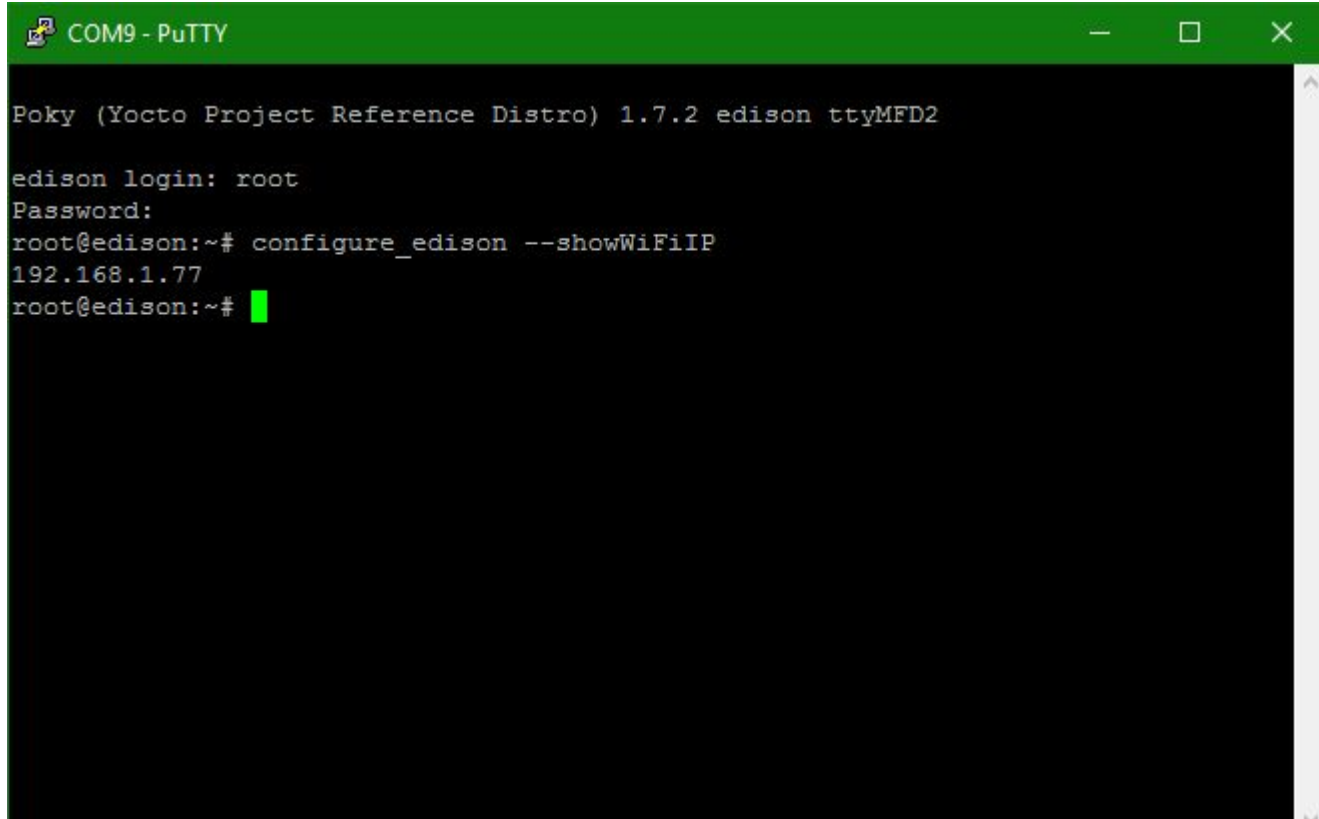
- IP Address
- username
- password
- Tag # OR Anchor #

SparkFun Receiver Blocks



Extreme reduction in size and power footprint

Why can't we SSH in when we know the IP?

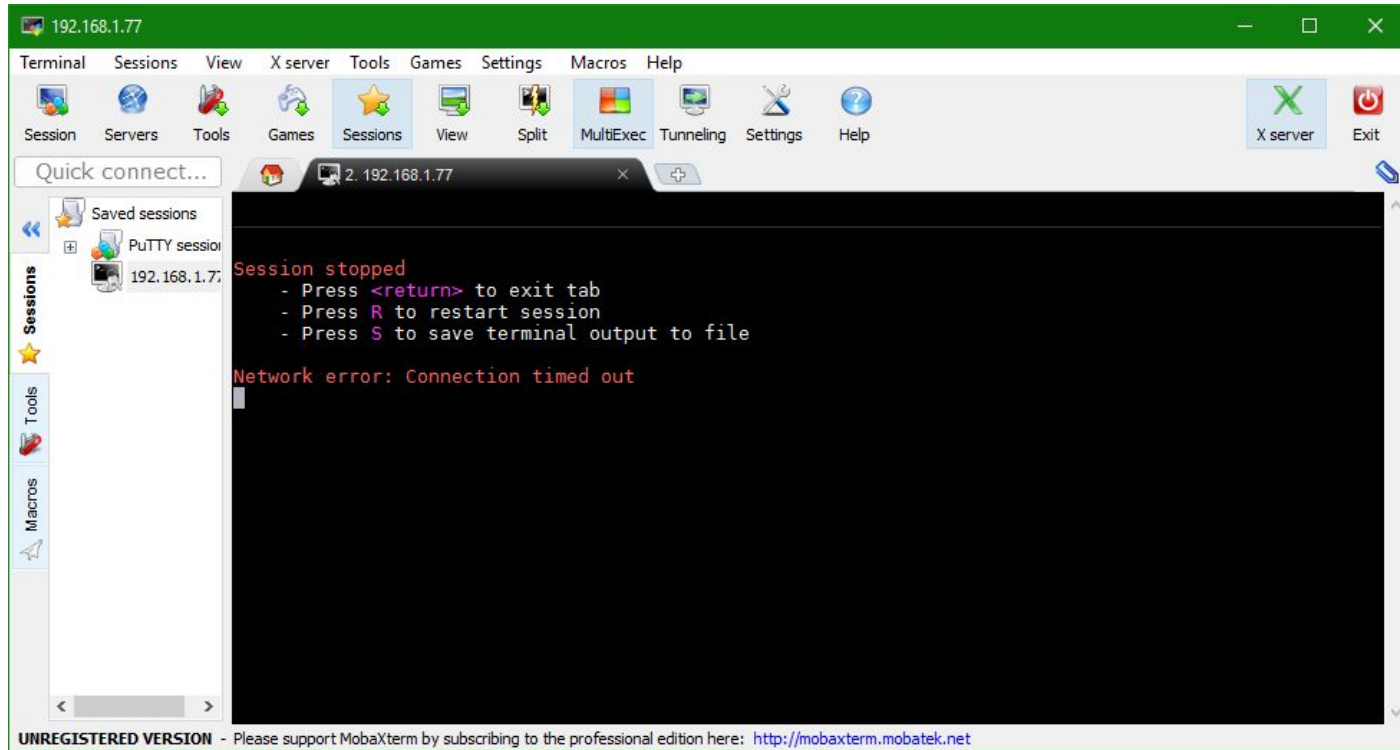


```
COM9 - PuTTY

Poky (Yocto Project Reference Distro) 1.7.2 edison ttyMFD2

edison login: root
Password:
root@edison:~# configure_edison --showWiFiIP
192.168.1.77
root@edison:~#
```

Accessing the SparkFun Blocks



Security Concerns

Concerns

1. **Replay Attack** : If someone records IR transmission , and replays it in some other part of the hospital
2. **Spurious IR emitters attack** : If an attacker emits 38Khz IR signal at different places in the hospital and disrupts the system
3. **Attacker deciphers code** : If an attacker figures out the way in which the preamble and message is sent

Solutions

1. **Intelligence by the Server** : The server can figure out if certain zones are consistently sending invalid locations and as a corrective action the hospital security is informed
2. **Variable length preamble** : Has been incorporated in the code. This involves a handshake between server and the smart tags

Comparison of our solution wrt. other technologies

Compare with UWB radio

Compare with Bluetooth /Zigbee

Any other possible technologies - Acoustic etc .

COST EFFICIENCY, POWER CONSUMPTION