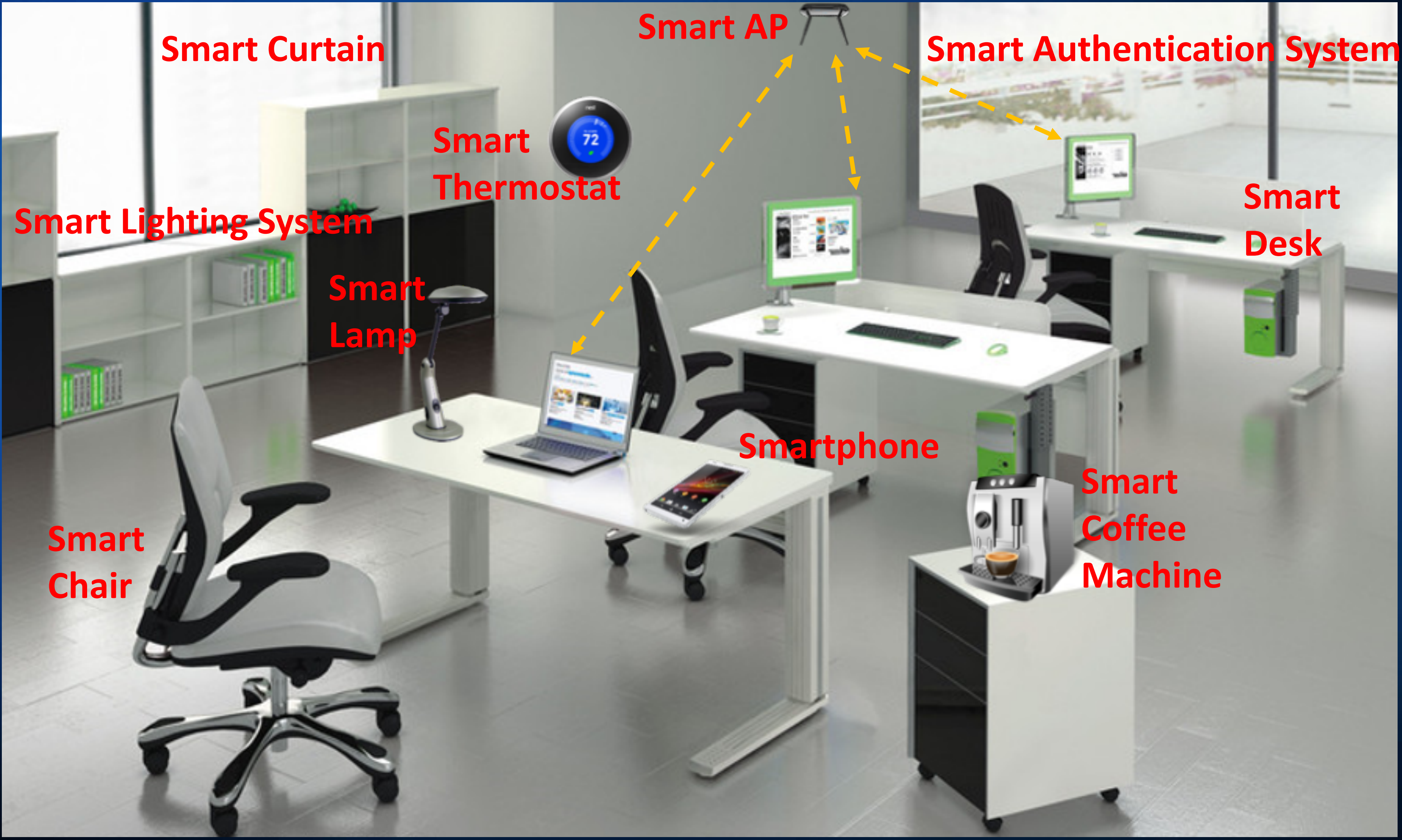
Security in Internet of Things

Prasant Mohapatra
Department of Computer Science
University of California, Davis

UCDAVIS
UNIVERSITY OF CALIFORNIA

IoT – around us



IoT – with us

Smart Glass

Temperature Sensor

Blood Pressure Sensor

Smartwatch

Smartphone

Heartbeat Sensor

Smart Luggage

WiFi-based Sensing

Shoe Sensors

Organization

- Introduction
- Secure Authentication
- Continuous Authentication
- Detection of Attacks
- Protection against Vulnerabilities
- Threats from Unconventional Sensing
- Visions for the Future

Secure Authentication

Authentication in Smartphones

- Authentication in smartphones
  - device unlock
  - app login
  - forum/website login

- Authentication types
  - Credential-based (User name / password)
    - What the user knows
    - Identity theft
    - Memory burden

Username : ******

Password : ******

Biometric Authentication

Voice

- **Inconvenient, vulnerable**
- Requires speaking, Background noise

Fingerprint

- **Convenient**, **vulnerable**
- **Expensive** hardware required
- **Limited market**

Face (and Iris)

- **Convenient**, **vulnerable**
- **Inexpensive** – Use mobile camera

Compelling. Let's explore further

Facial Authentication

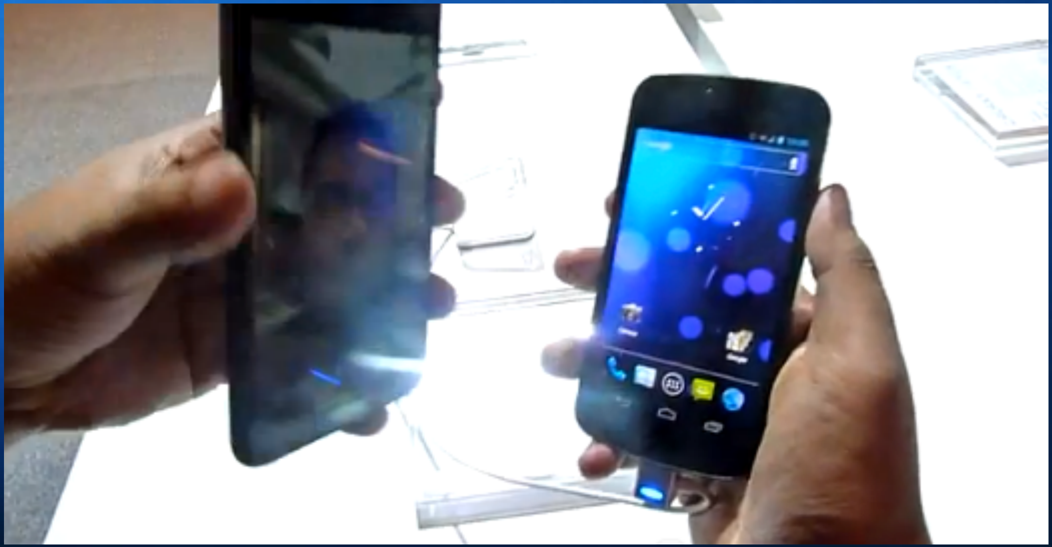
- Face verification / face identification
- Face recognition accuracy has been largely improved
  - Accuracy is very close to 100%
  - Even used for commercial payment systems
- Most smartphones have front-facing cameras; usually higher than 10 M pixels
  - Convenient for face capturing
  - Quality is good enough for face recognition

Current Status

- Android: face unlock alternative since 4.0
  - But not many users are using it
- App and website login
  - User name / password dominates other methods
- Why facial authentication is not widely used in smartphones?
  - Privacy concerns
  - Security issues
    - 2D media attacks
    - Virtual camera attacks
  - usability

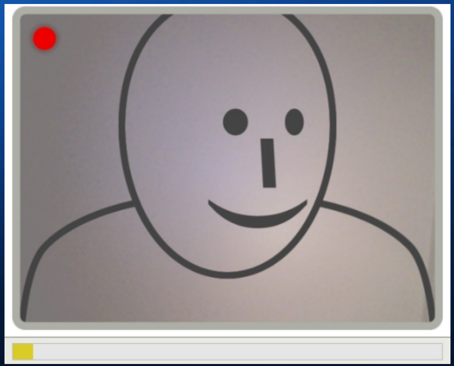
2D Media Attack

- Photo attack (print attack)
  - Use user’s photo to cheat the authentication system
- Video attack
  - Starting from Android 4.1, eye-blink is required
  - use video to compromise the system



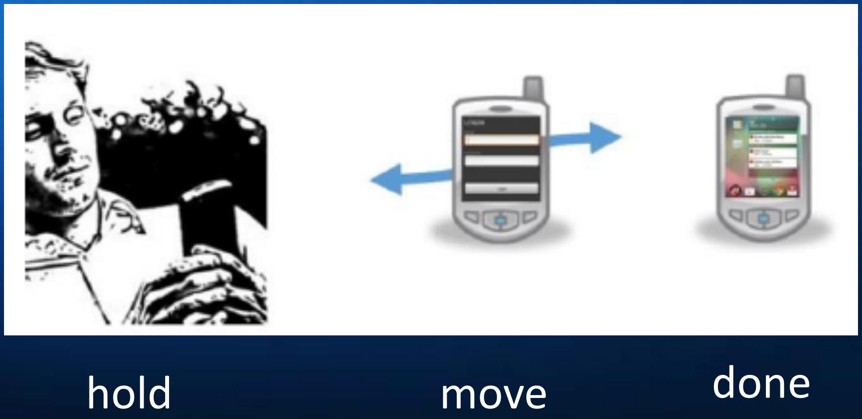
2D Media Attack (cont.)

- 3D facial recognition can defend against this attack
  - 3D template matching
  - e.g. Toshiba Face Recognition Utility
- Difficult to use
- Turning heads towards different directions -> user's burden
  - A trial takes more than 20 seconds -> much longer than entering password
  - Even a genuine user may need multiple trials to pass



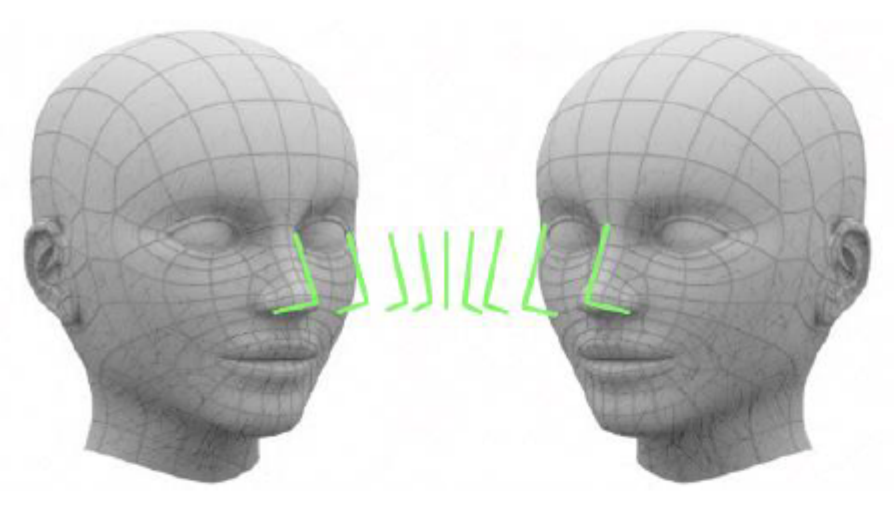
Our Method

- Achieve high security and usability simultaneously
  - Safe for 2D media attacks
  - Safe for virtual camera attacks
  - Much faster than 3D face authentication method (speed is comparable to credential-based method)： ~2 sec
- How?
  - Only need to move the phone in front of face for a short distance
  - Utilizing motion sensors in smartphones
  - No need to move head and sync with directions



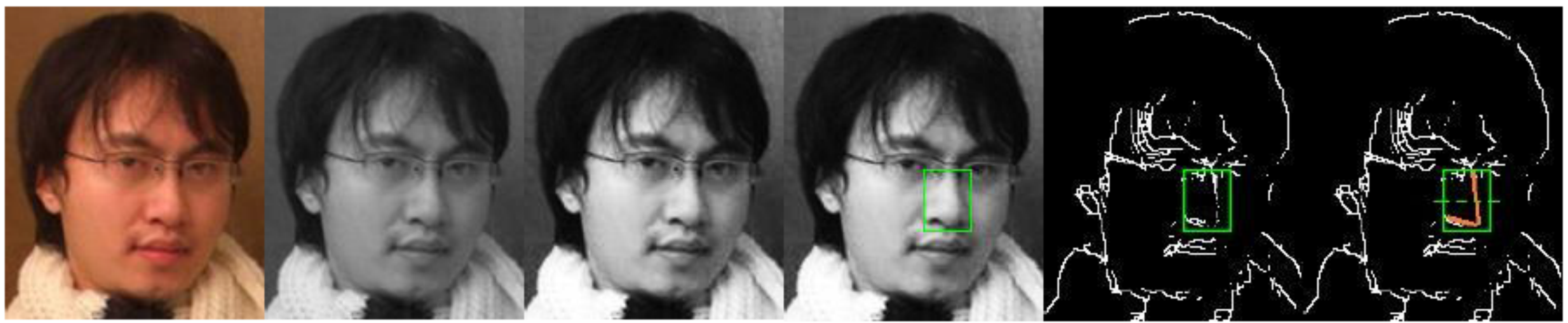
Counter 2D Media Attack

- Idea
  - Nose orientation changes when moving phone horizontally if a real 3D face



Nose Angle Detection

- Detect nose outline
  - Video frame preprocessing
  - Nose detection (can employ existing method)
  - Nose outline fitting



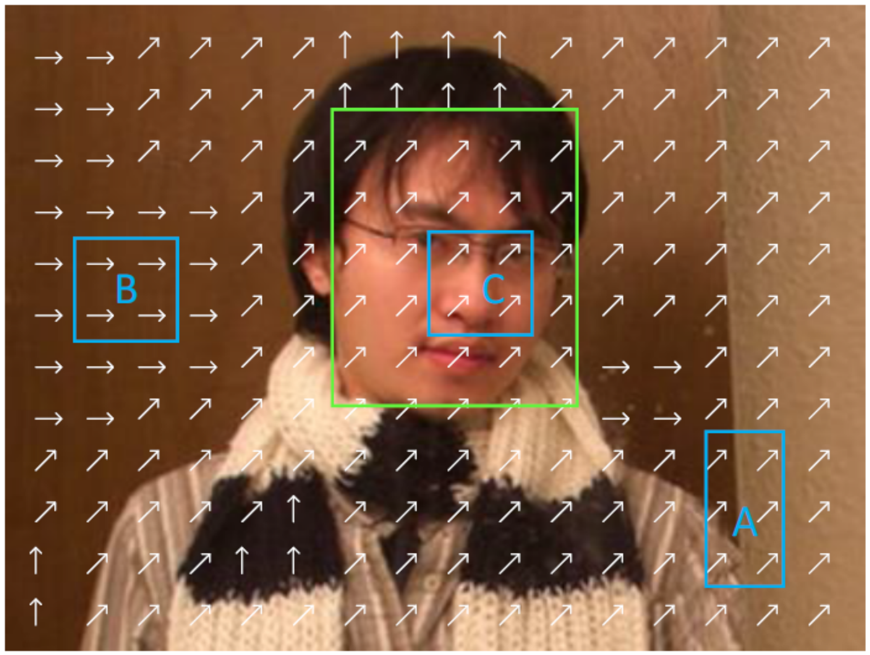
- Compare nose outline from two sides
  - Motion sensors: judge the relative position between face and smartphone, picking correct frame intelligently
  - Light sensor: auto boost screen brightness if dark, to enhance luminance (improve nose outline detection)

Counter Virtual Camera Attack

- Idea !
  - If real-time video captured by physical cam, small shakes in video should be consistent with smartphone’s motion sensor readings
  - Pre-recorded videos can be detected
  - Assume motion sensor readings are not compromised

Motion Vector Correlation

- Small motions extracted from the video

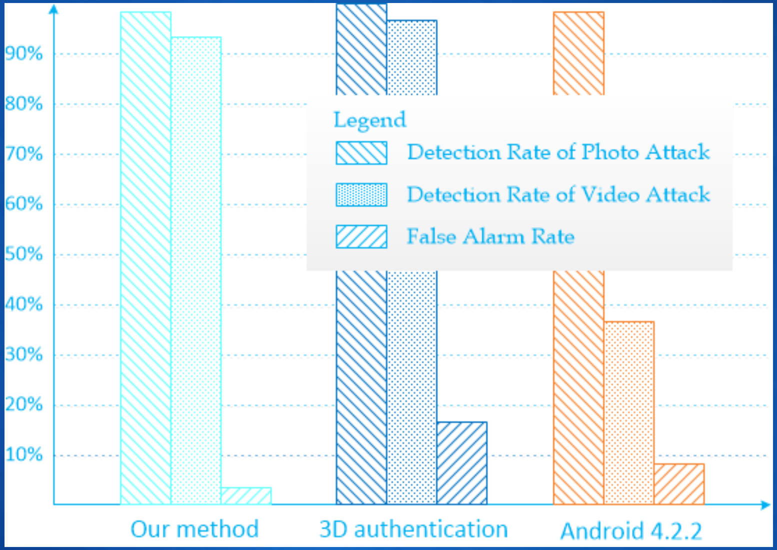


- Compare with small shakes extracted from motion sensors

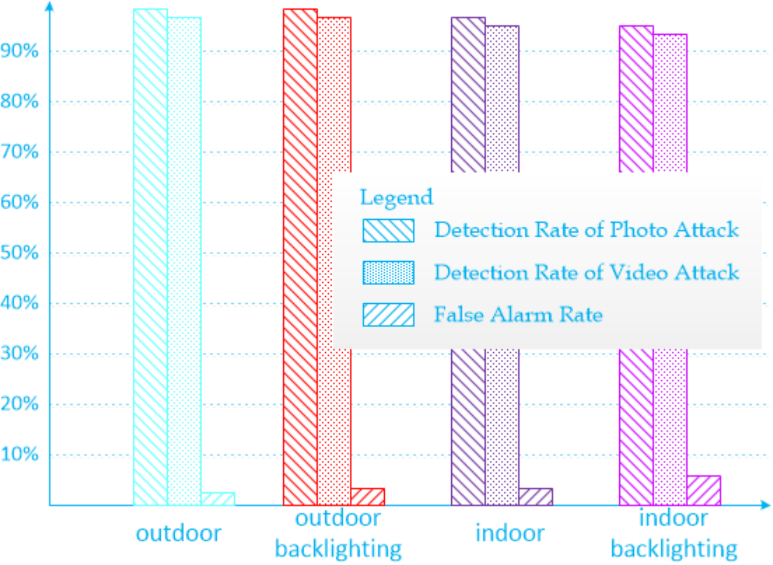
Evaluations

- Samsung Galaxy Nexus with 1.3M pixel front-facing camera
- Android 4.2.2
- Video is 480*720@24fps, chopped to 480*640
- Use *Haar Cascades* in OpenCV to detect face and nose
- Face recognition algorithms are orthogonal to our method, but for completeness, we do include a PCA (principal component analysis) based facial identification module (also implemented using OpenCV)

Accuracy of 2D Media Attack Detection (cont.)

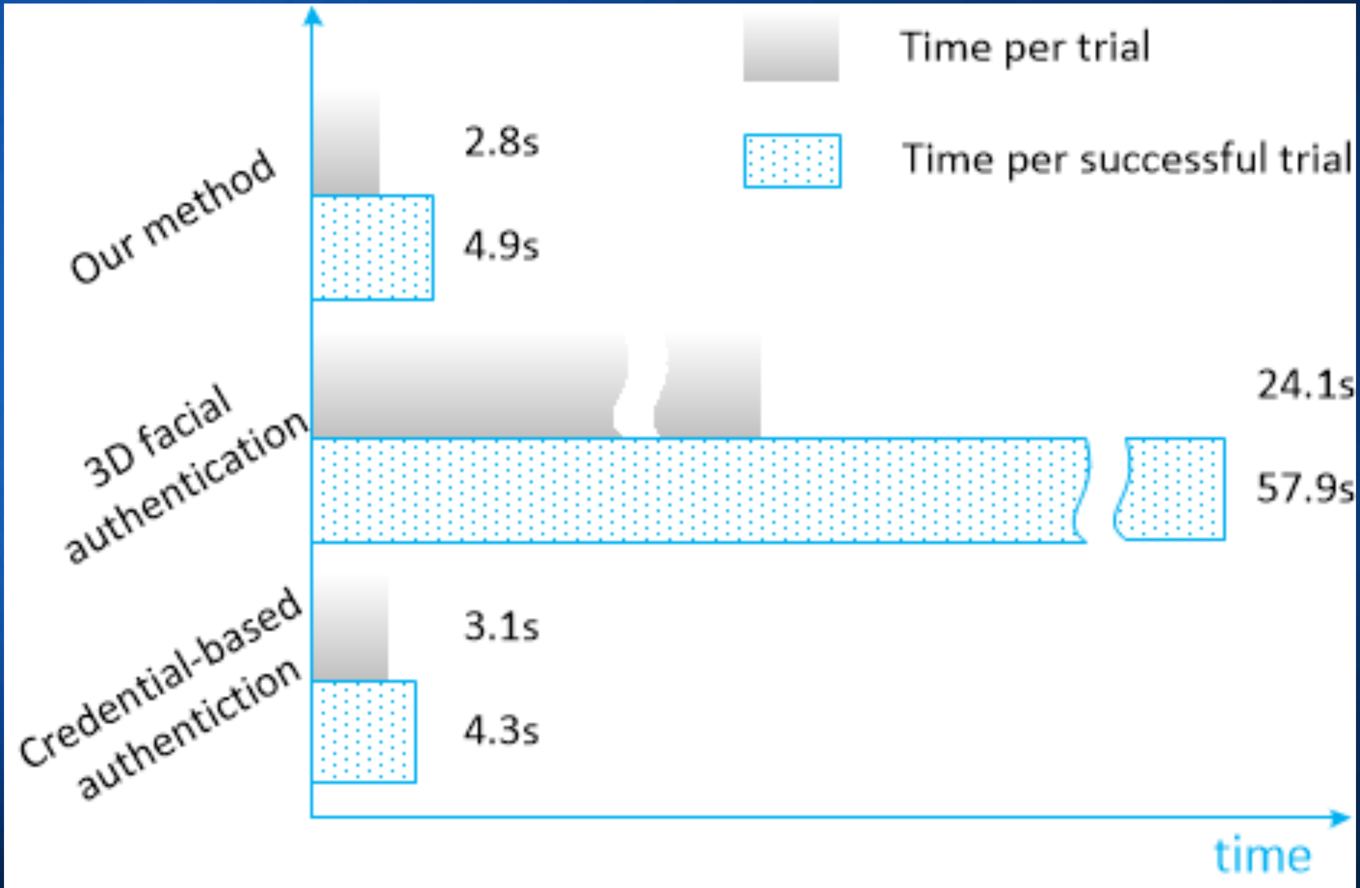


Accuracy compared with other state-of-art approaches



Accuracy under different illuminance

Authentication Time



PCASA: PROXIMITY-BASED CONTINUOUS AND SECURE AUTHENTICATION

Motivations for PCASA:

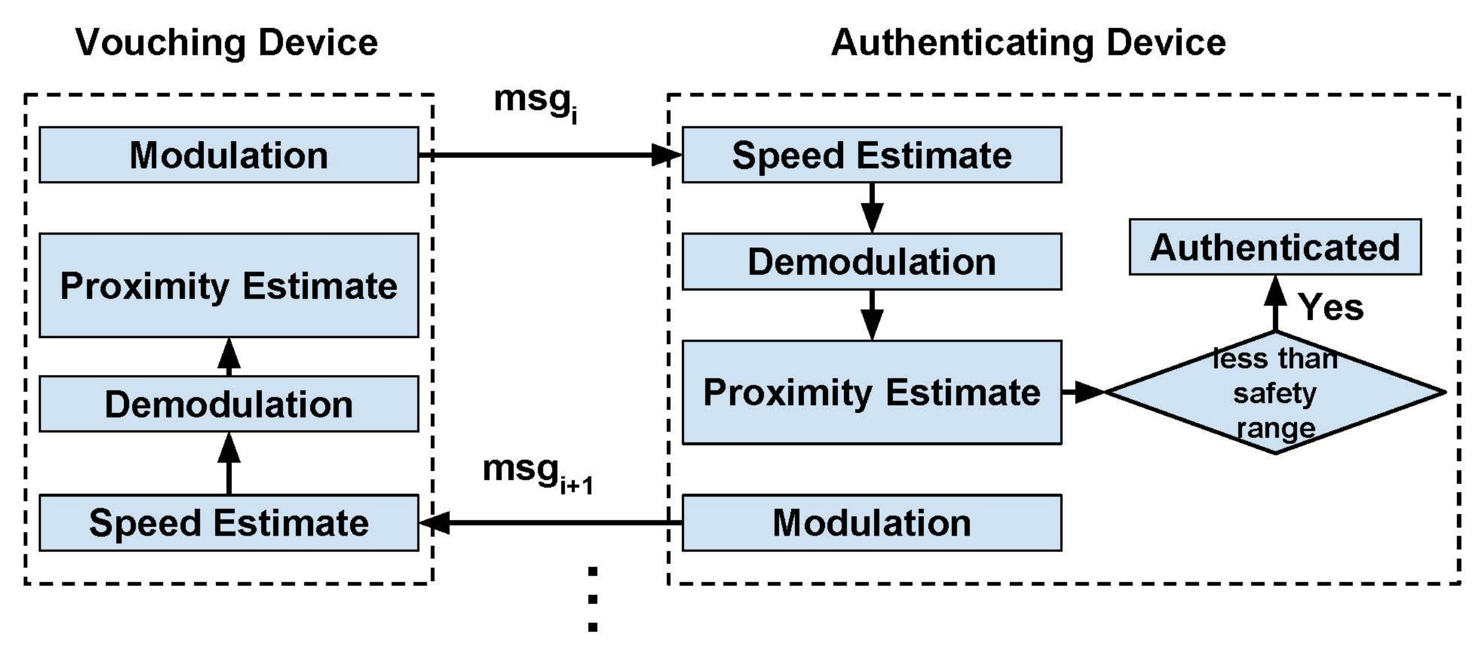
- Leveraging devices that are within user's vicinity and physical control
- Enable continuous authentication
- Easy to authenticate multiple devices
- Challenges:
  - RF technology unable to measure the proximity at sub-meter level due to large fluctuation of signal.
  - Current acoustic based approach:
    1. unable to exchange credential information among devices.
    2. unable to handle energy efficiency problem.

PCASA: Objective

- **Security:** Defend against the attackers who aim to get illegitimate access to user’s portable device
- **Accuracy:** Estimate the proximity with sub-meter accuracy in real-time even when the user is mobile
- **Energy Efficiency:** Perform continuous authentication using acoustic signals with low energy.

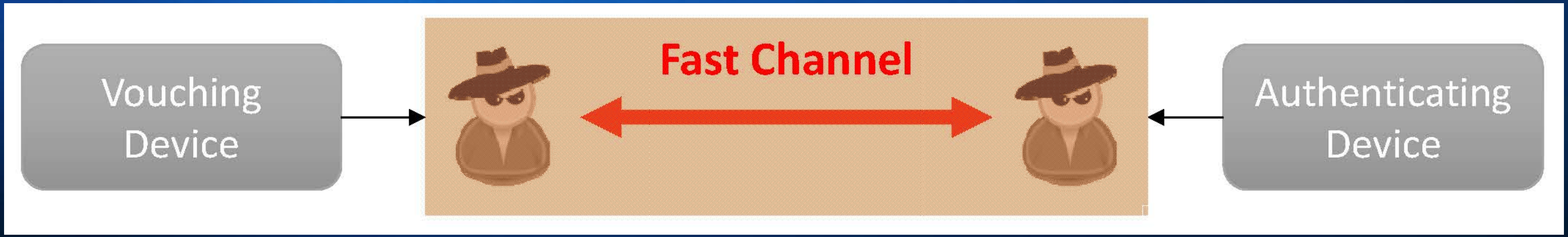
PCASA: System Overview

- Vouching Device: wearable device always on the body (e.g. smartwatch, glasses)
- Authenticating Device: portable devices not always on body (e.g. laptop, tablet)



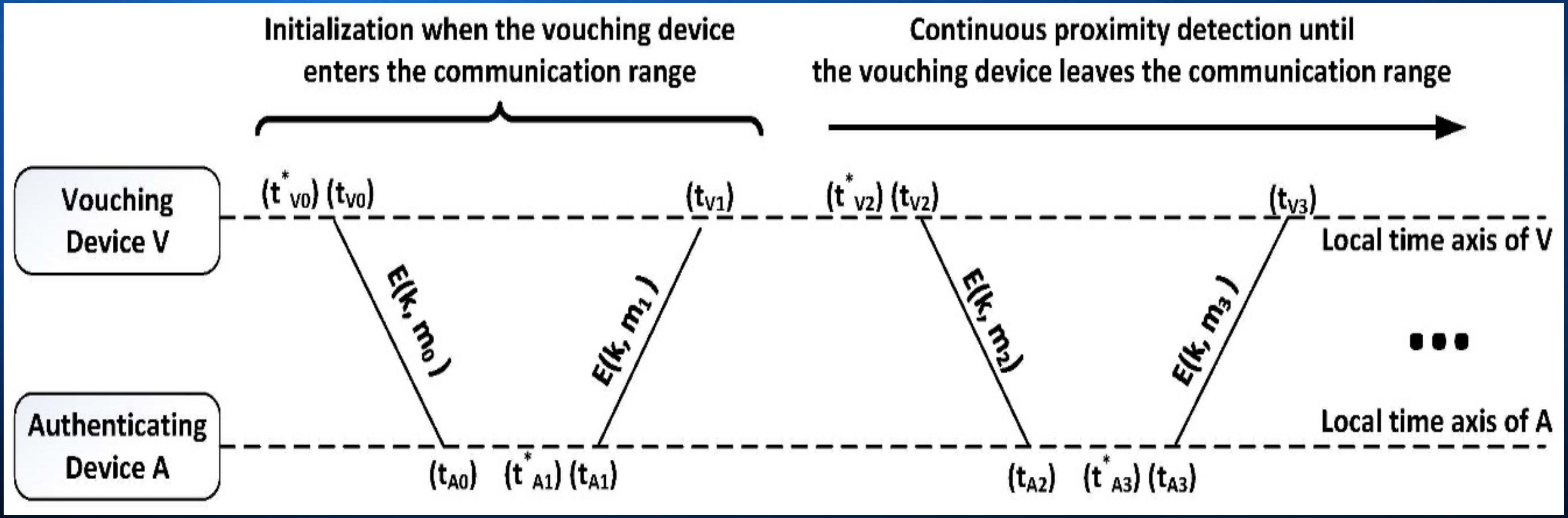
PCASA: Attack Model

- Zero-Effort Attacks
  - Directly access the authenticating device while out of user’s vicinity or control.
  - Exist in RF based approaches.
- Spoofing Attacks
  - Replay Attacks: replays the recorded signal from a short distance to spoof the authenticating device
  - Relay Attacks: create a faster channel to relay all messages between the vouching and authenticating devices



PCASA: Protocol - Initialization

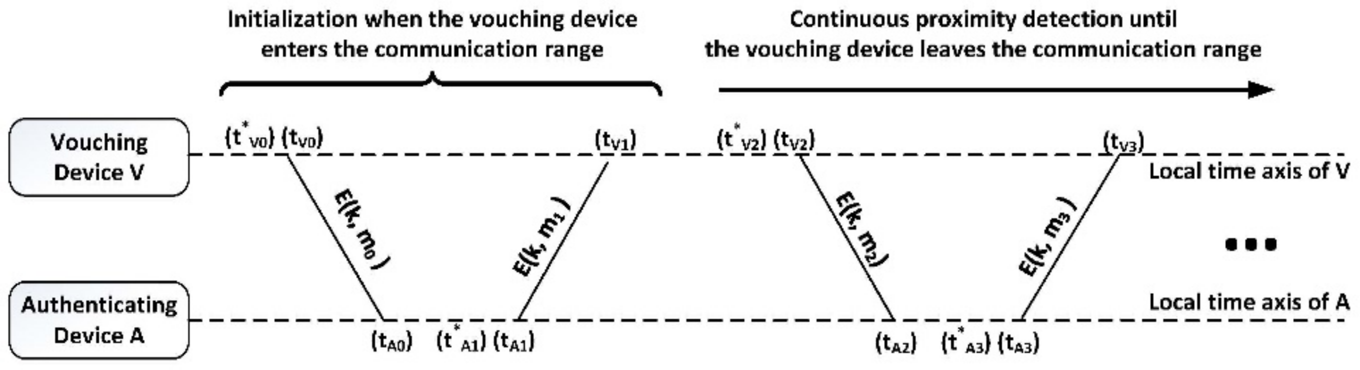
- Initialization
  - Vouching sends message m_0 to authenticating at t_{v0}, where m_0 contains MAC address of vouching.
  - Authenticating receives m_0 at t_{A0}
  - Authenticating sends m_1 to vouching at t_{A1}, where m_1 contains MAC address of authenticating.
  - Vouching receives m_1 at t_{V1}


Initialization when the vouching device enters the communication range
Continuous proximity detection until the vouching device leaves the communication range
Vouching Device V
Authenticating Device A
(t^*_{V0}) (t_{V0})
(t_{V1})
(t^*_{V2}) (t_{V2})
(t_{V3})
Local time axis of V
$E(k, m_0)$
$E(k, m_1)$
$E(k, m_2)$
$E(k, m_3)$
Local time axis of A
(t_{A0}) (t^*_{A1}) (t_{A1})
(t_{A2}) (t^*_{A3}) (t_{A3})

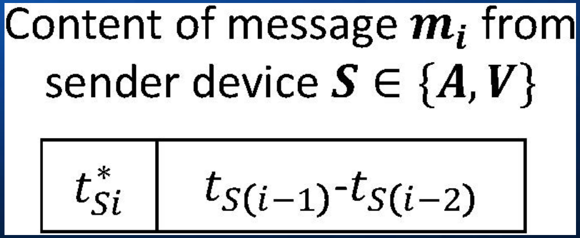
PCASA: Protocol - Continuous Proximity Detection

- Continuous Proximity Detection
  - Vouching device sends message $m_2$to authenticating at t_{V2}.
  - Authenticating device receives m_2 at t_{V2}
  - Authenticating device calculates its distance to vouching as follows,

$$\frac{c}{2}[(t_{V1} - t_{V0}) - (t_{A1} - t_{A0})]$$



(a) Protocol Overview



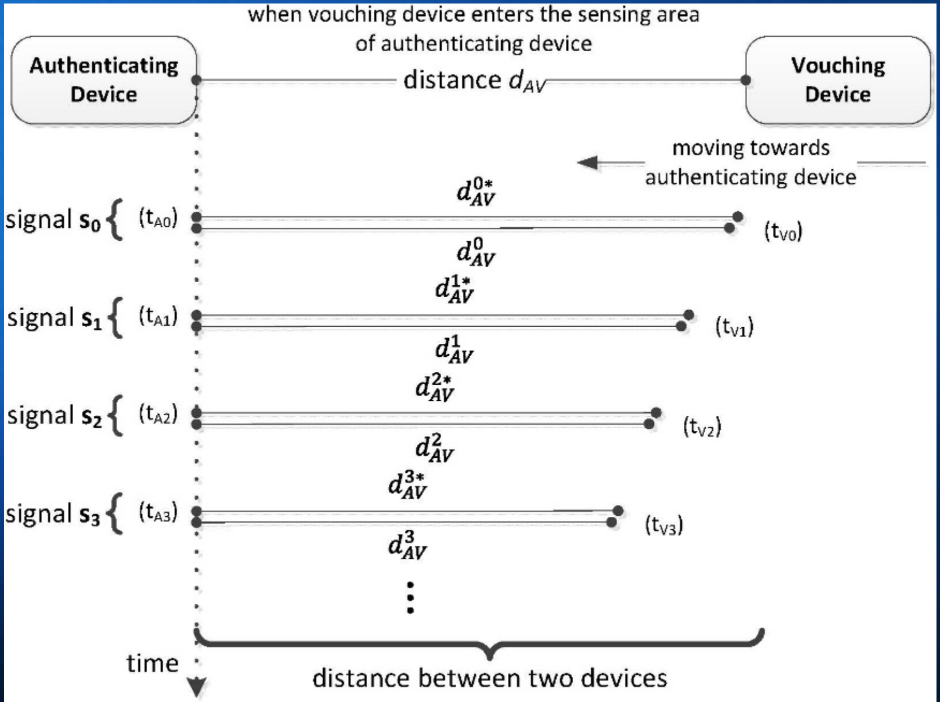
Content of message $\boldsymbol{m_i}$ from sender device $\boldsymbol{S} \in \{\boldsymbol{A}, \boldsymbol{V}\}$

t^*_{Si}	$t_{S(i-1)}$-$t_{S(i-2)}$

(b) Messages

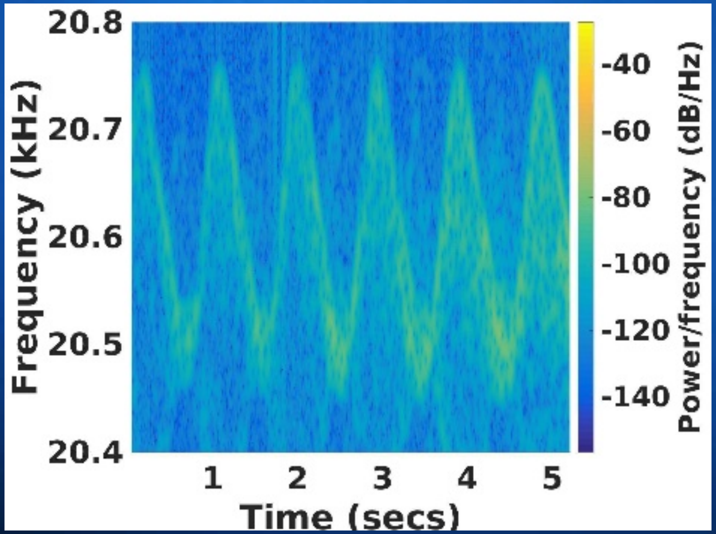
PCASA: Protocol - User Mobility

- Measure the proximity when moving
  - Calculate distance d_{AV}^{0} and d_{AV}^{1}
    - $d_{AV}^{1} - d_{AV}^{0} = v(t_{V1} - t_{V0})$
    - $d_{AV}^{1} + d_{AV}^{0} = c[(t_{V1} - t_{V0}) - (t_{A1} - t_{A0})]$
  - Calculate d_{AV}^{2} based on d_{AV}^{1}
    - $d_{AV}^{2} = d_{AV}^{1} - v(t_{V2} - t_{V1})$

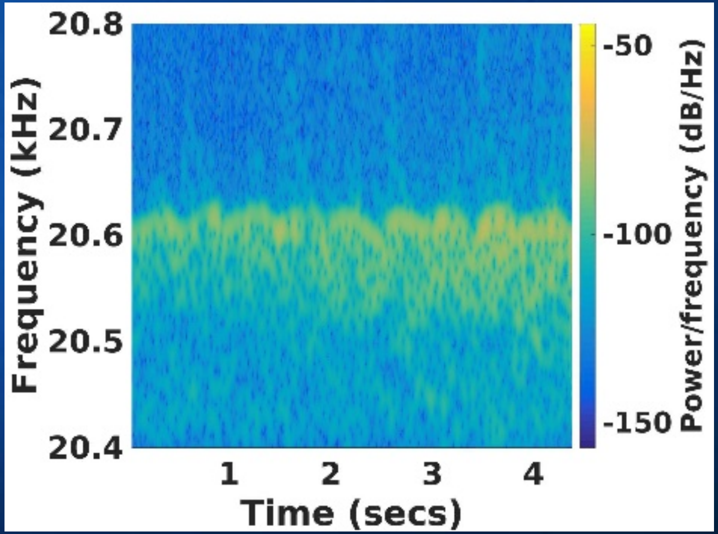


User Mobility – Estimate speed

- Measure the Relative Speed using Doppler Effect
  - Doppler Effect: f $= \frac{v}{v_a} f_0$
  - Example: sound at 20kHz, 1Hz shift corresponds to $\frac{1*340m/s}{20k} = 0.017m/s = 1.7cm/s$
  - two scenarios of human walk: 1) in pocket, 2) on hand



(a) Spectrogram – watch on hand



(b) Spectrogram – phone in pocket

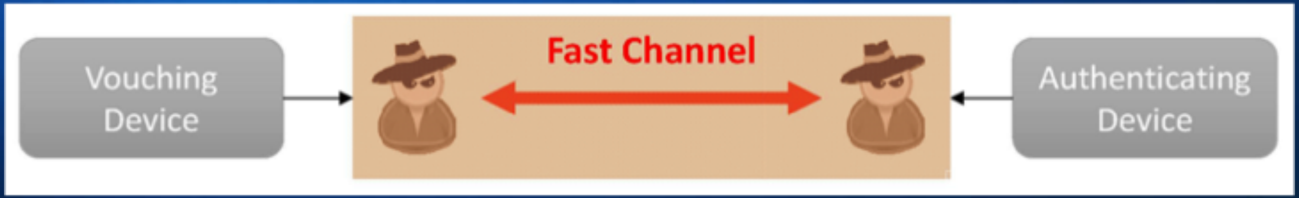
Security Analysis

- Zero-Effect Attacks can be defended as distance between vouching and authenticating can be accurately measured
- Replay Attacks will delay the message, leading to a larger arrival time, i.e. larger distance.

$$d_{AV}^{1} + d_{AV}^{0} = c[(t_{V1} - t_{V0}) - (t_{A1} - t_{A0})]$$

- Relay Attack is impossible without attracting user's attention.

Relay Attacks

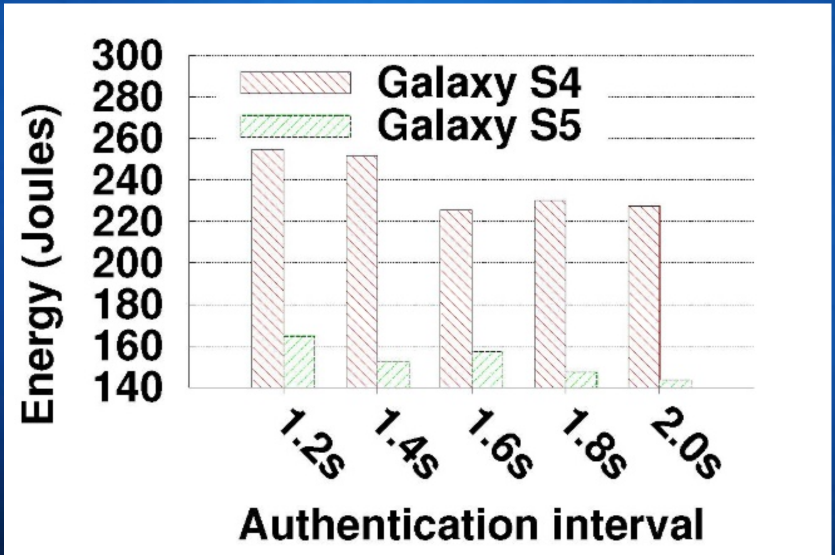


Evaluation:
Experiment Setup and Implementation

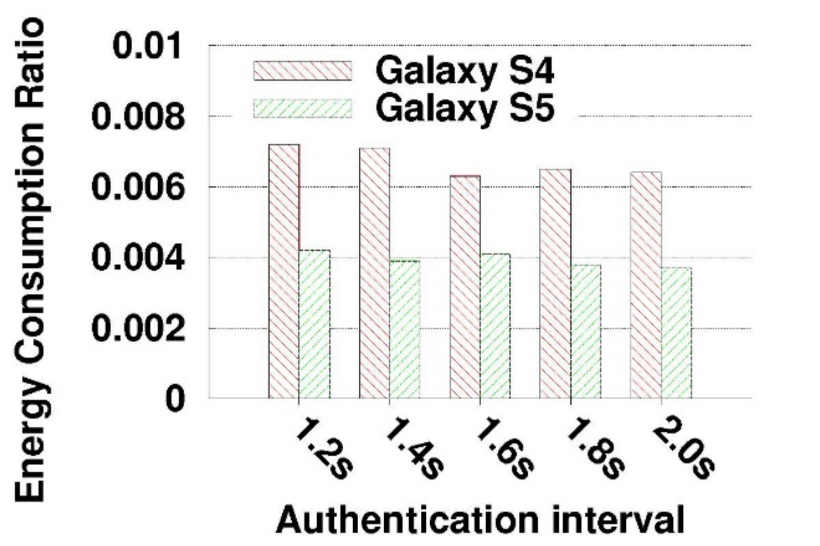
- Devices: Samsung Galaxy S4(1), Samsung Galaxy S5(2), Samsung Galaxy S6(1), iPhone 6S(1), Apple Watch(1), Samsung Gear S2-LTE(1).
- Acoustic signal is generated at 20 kHz and speaker is set at the highest volume
- Sampling rate of the microphone for recording is set to 44.1kHz

Evaluation: Energy Consumption

- For the most energy consuming device – Galaxy S4, it could perform continuous authentication for up to 34 hours with the highest authentication rate



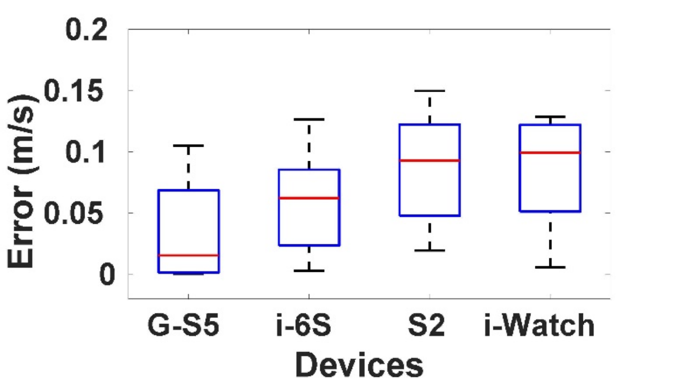
(a) Energy consumption at different authentication rates



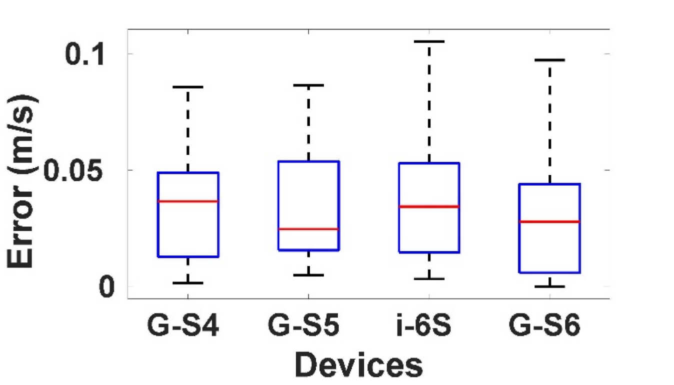
(b) Energy Consumption Ratio

Evaluation: Speed Estimate

- Devices on hand have relatively higher estimate error than devices in the pocket.
- Error on all devices in our experiments does not exceed 0.15 m/s



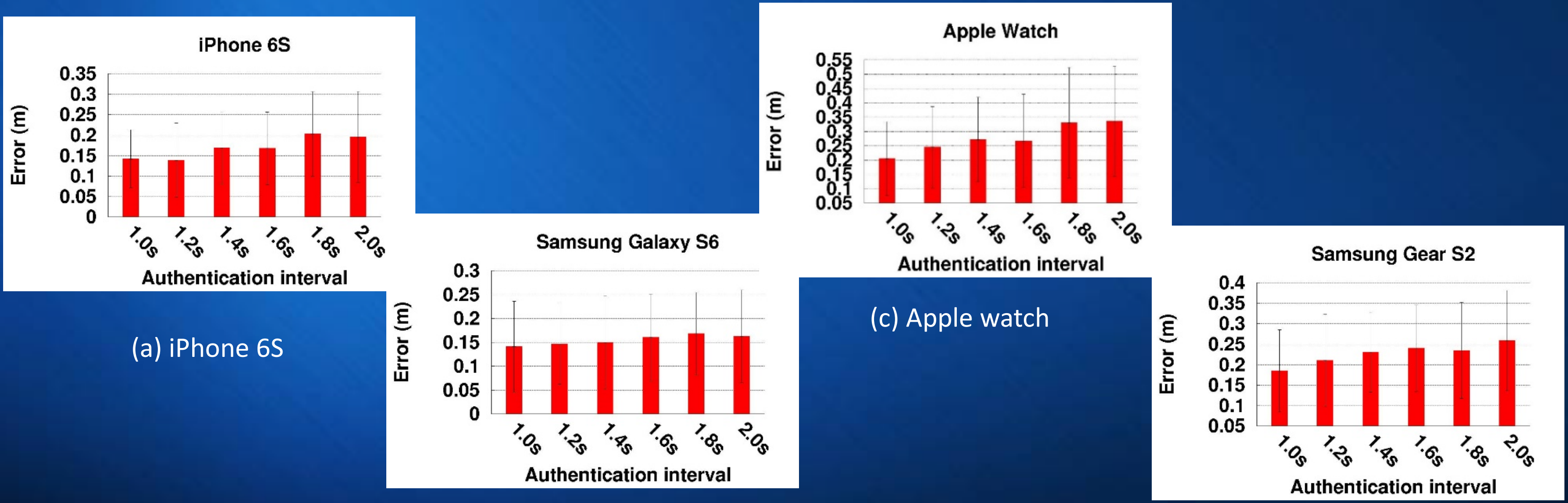
(a) Device worn on wrist



(b) Device carried in pocket

Evaluation: Proximity Estimation

- Proximity estimation error increases along with the authentication interval as proximity estimate is related with speed estimate and message interval/authentication frequency
- Average error of proximity estimation is no more than 0.25m even when the user is mobile



(a) iPhone 6S

(b) Galaxy S6

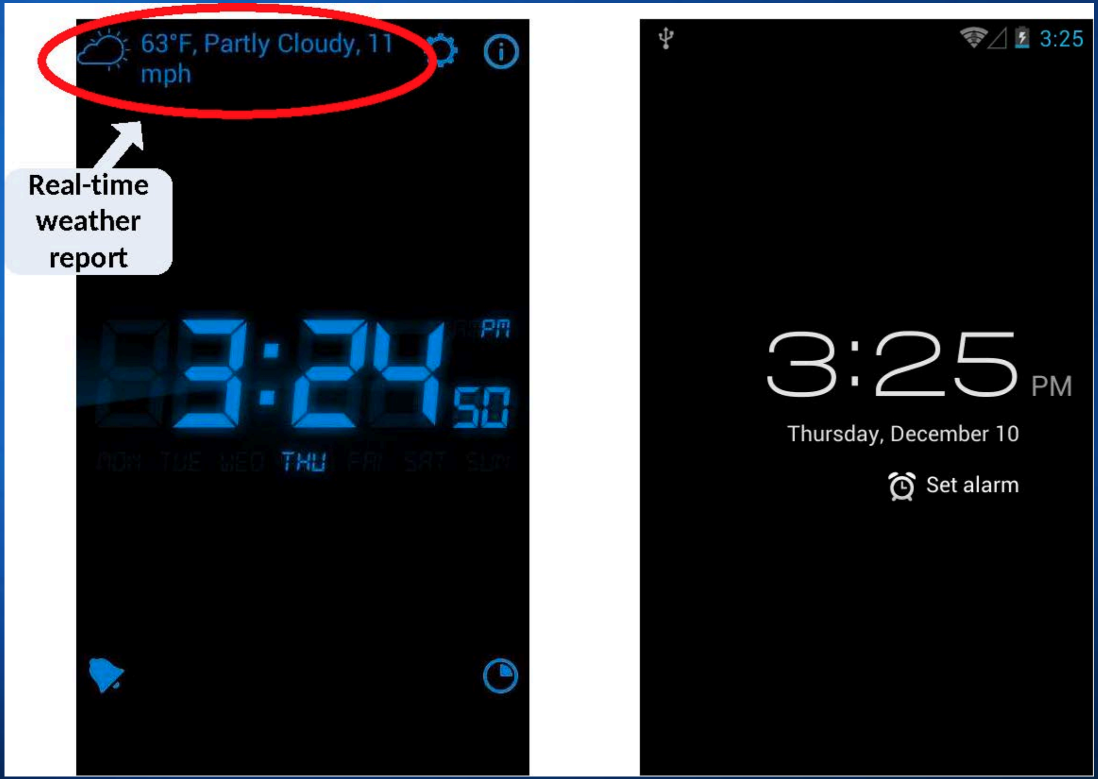
(c) Apple watch

(d) Gear S2

FlowIntent:
Detecting Suspicious Apps

Motivating Example

- User interfaces of Two Clock apps



Real-time weather report

Motivating Example

- Both apps send out user location through HTTP traffic under shown user interfaces
  - Legal for the first app while suspicious for the second
  - Need to understand user intention
- Vulnerability in Android permission control system
  - Mismatch between user intention and app behavior
- Standard approaches
  - Dynamic and static program analysis
  - High overhead at host end
  - Early stage on user intention modeling

Network based Detection

Objective

- Detect suspicious behavior only from network traffic data
- Incorporate user intention to improve accuracy

Advantages

- Low overhead at host: easy to deploy at IDS or access point
- Monitor a large number of devices without introducing overhead at the end hosts, update-to-date signatures
- Signatures not revealed by system-level approaches

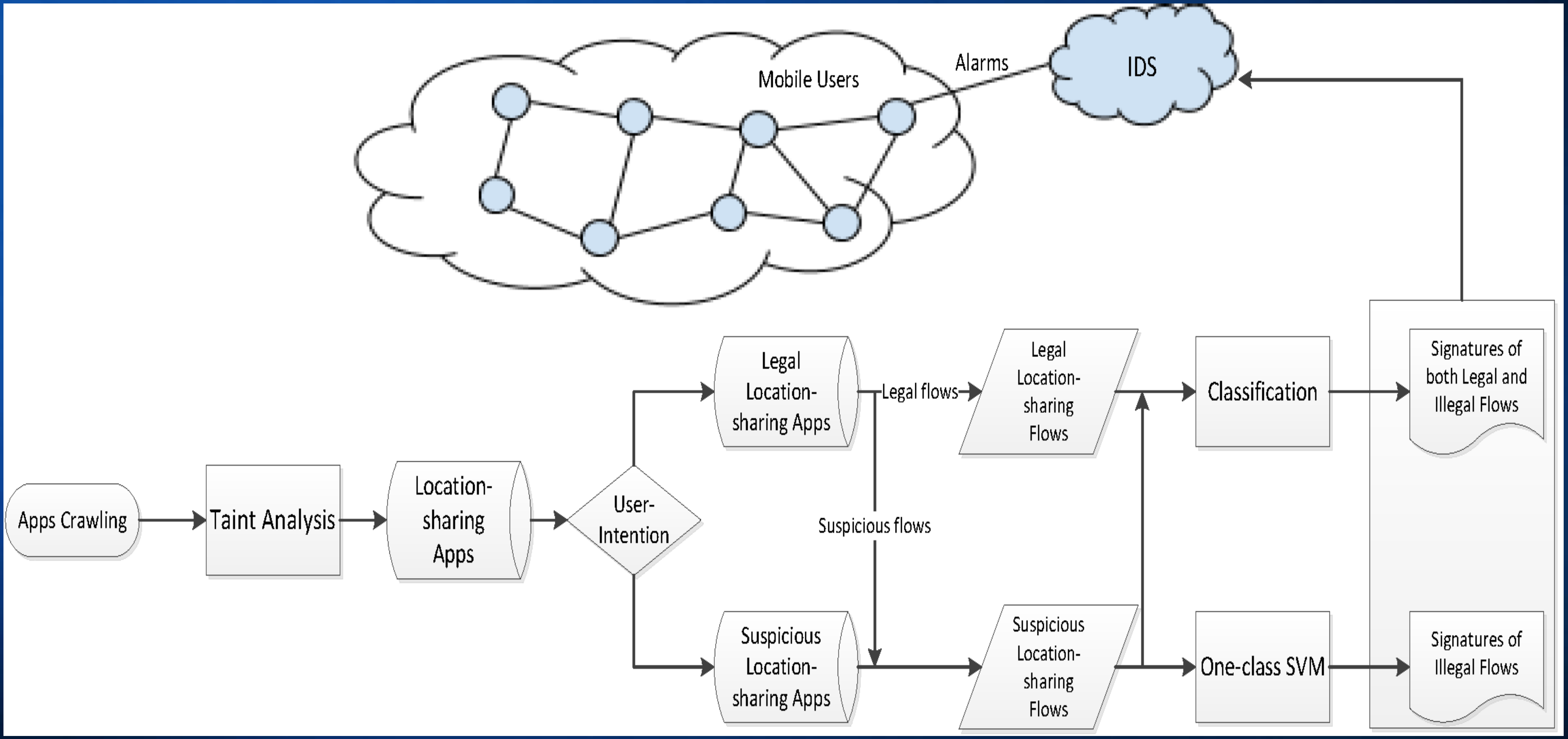
Feasibility

- Most suspicious traffic are transmitted with simple unencrypted HTTP requests
- Number of malware families are not huge
- Variants of the same malware exhibit similar behaviors

Approach Overview

- Dataset
  - Automatically run apps and collect their network traffic
  - Identify location-sharing apps with taint analysis
  - 1268 location sharing apps identified from 20,000 apps crawled from Google Play and Baidu App Market
- User Intention modeling
  - Features: app name, description, and user interface.
  - all location transmissions from suspicious apps are marked as suspicious.
  - legal apps may also generate unintended flows: some can be identified from existing black list.
- Machine learning on app traffic flows
  - statistical features and lexical features
  - Only network level features are used in testing

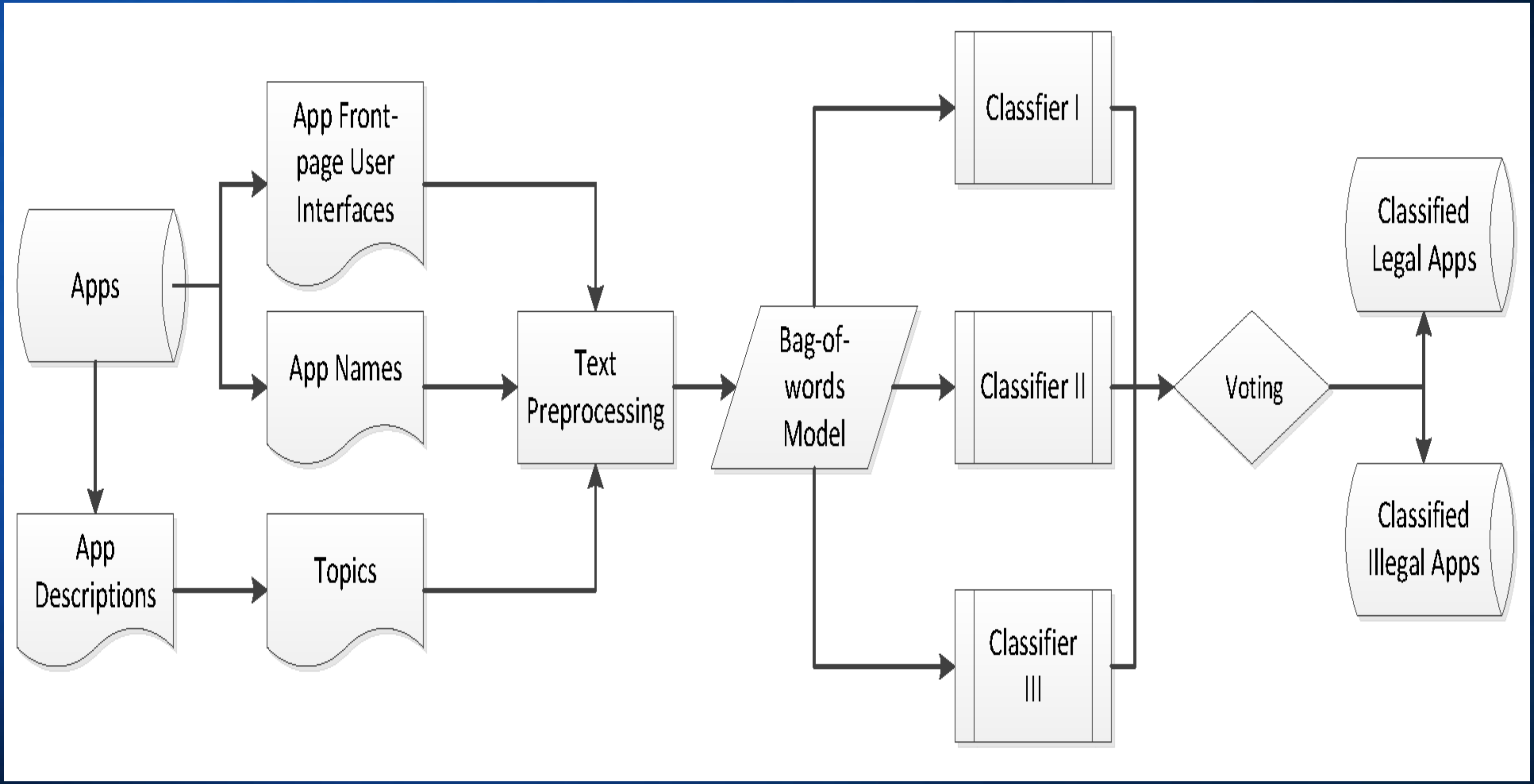
System Architecture


Mobile Users
Alarms
IDS
Apps Crawling
Taint Analysis
Location-sharing Apps
User-Intention
Legal Location-sharing Apps
Legal flows
Legal Location-sharing Flows
Classification
Signatures of both Legal and Illegal Flows
Suspicious flows
Suspicious Location-sharing Apps
Suspicious Location-sharing Flows
One-class SVM
Signatures of Illegal Flows

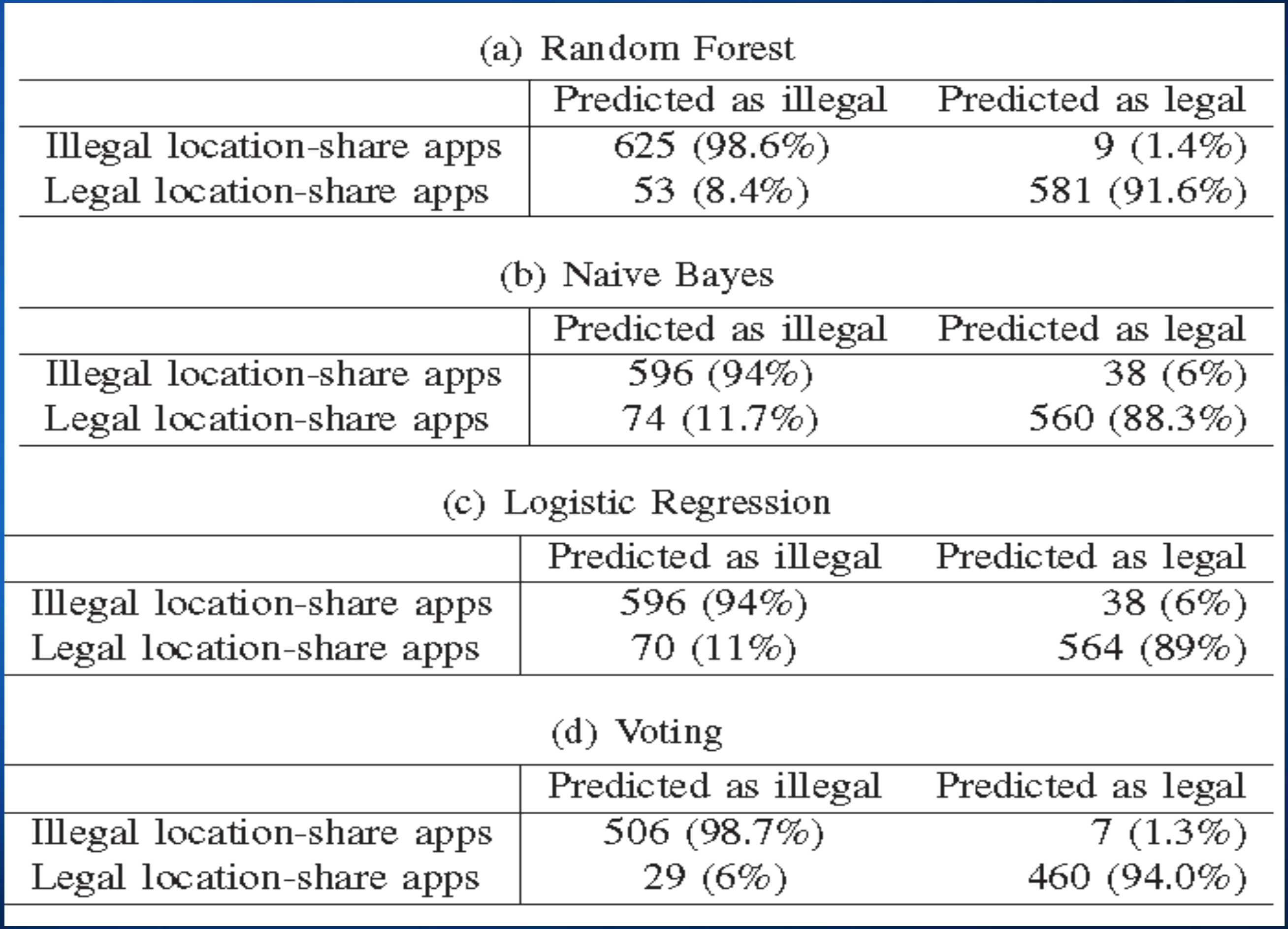
User Intention Modeling

- Model user intention from text features and GUI data
  - app names, app topics
  - user interfaces: currently focus on front-page UI and traffic flows under that UI
  - Leverage NLP and bag-of-words to extract text features
- Evaluate classification results through 10-fold cross validation
- Better accuracy via multiple classifiers and voting

User Intention Modeling



User Intention Modeling: Results



(a) Random Forest

	Predicted as illegal	Predicted as legal
Illegal location-share apps	625 (98.6%)	9 (1.4%)
Legal location-share apps	53 (8.4%)	581 (91.6%)

(b) Naive Bayes

	Predicted as illegal	Predicted as legal
Illegal location-share apps	596 (94%)	38 (6%)
Legal location-share apps	74 (11.7%)	560 (88.3%)

(c) Logistic Regression

	Predicted as illegal	Predicted as legal
Illegal location-share apps	596 (94%)	38 (6%)
Legal location-share apps	70 (11%)	564 (89%)

(d) Voting

	Predicted as illegal	Predicted as legal
Illegal location-share apps	506 (98.7%)	7 (1.3%)
Legal location-share apps	29 (6%)	460 (94.0%)

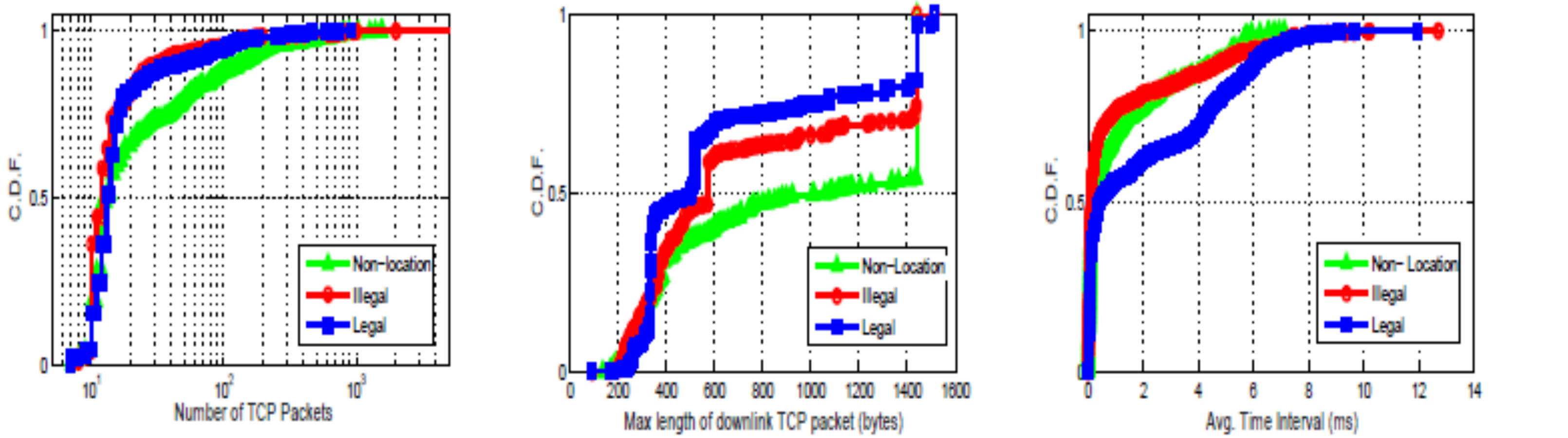
Network-level Features

- Statistical Features
  - Total number of TCP packets
  - Total number of uplink TCP packets
  - Total number of HTTP packets (Packets with HTTP application layer present)
  - Packet size of all TCP packets
  - Packet size of uplink TCP packets
  - Packet size of downlink TCP packets
  - Time interval between two consecutive TCP packets
- Lexical Features
  - Binary feature for each token in the host name and in the path URL
  - Length of the host name and entire URL
  - Number of dots in the URL

Statistical Features

- Packet Size: Ad flows usually respond with an advertisement with larger downlink packet size than legal flows

- Time Interval: packets are sent throughout the app usage in non-location flows, but like a burst in illegal flows



(a) Total number of TCP Packets

(b) Max length of downlink TCP packets (bytes)

(c) Time interval between two consecutive TCP packets (ms)

Fig. 5. C.D.F.s of Statistical Attributes

Lexical Features

- Illegal usage of location flow
  - ads.appsgeyser.com/?&guid=a5141e1d&tlat=38.53203&tlon=-121.759603&p=android&test=1
  - “ads” prefix indicates the advertisement purpose of the request.
- Location-sharing flow generated by a weather forecast application begins with the URL
  - v.juhe.cn/weather/geo?&lon=-121.750683&lat=38.540323
  - “weather” suggests the server behind the URL is a weather information provider

Traffic Classification

- All location flows generated by suspicious apps are marked as suspicious.
- Utilize existing bad host names list to remove suspicious flows generated by legal apps, and label rest flows as benign.
  - domain names of malware, ad and analytics servers
- Achieves a precision of 91.3% by using both statistical and lexical features. When true app classes are used, the precision increases to 92.8%.
  - Ground truth for unencrypted flows: manually check URL and plain text inside payload
  - Ground truth for encrypted flows: use firewall to block flows and examine its effect on app behavior
  - 10-fold cross validation
  - Our user intention modeling only incurs a slight loss in accuracy, while saving the effort of manually labeling a large number of apps

iType:
Using Eye Gaze to Enhance Typing Privacy

Wearables

- Accelerometers
- Gyroscope
- Ambient light sensor
- Hart rate sensor
- Magnetometer
- GPS
- …

Extend beyond timing → **daily life**

fitness

[1] https://www.iphones.ru/wp-content/uploads/2015/05/main.jpg

However

Explicitly typing sensitive info.

- Password
- Personal data
- Security code
- ….

Continuously sense hand moves

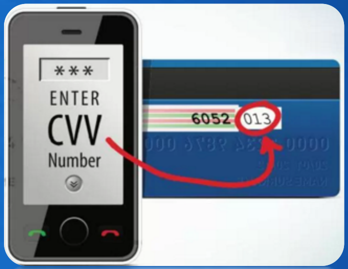
- Accelerometers
- Gyroscope
- ….

Wait a moment ...

- Touch ID

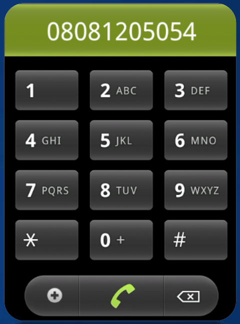
- But

Account login



Security code

POS terminal

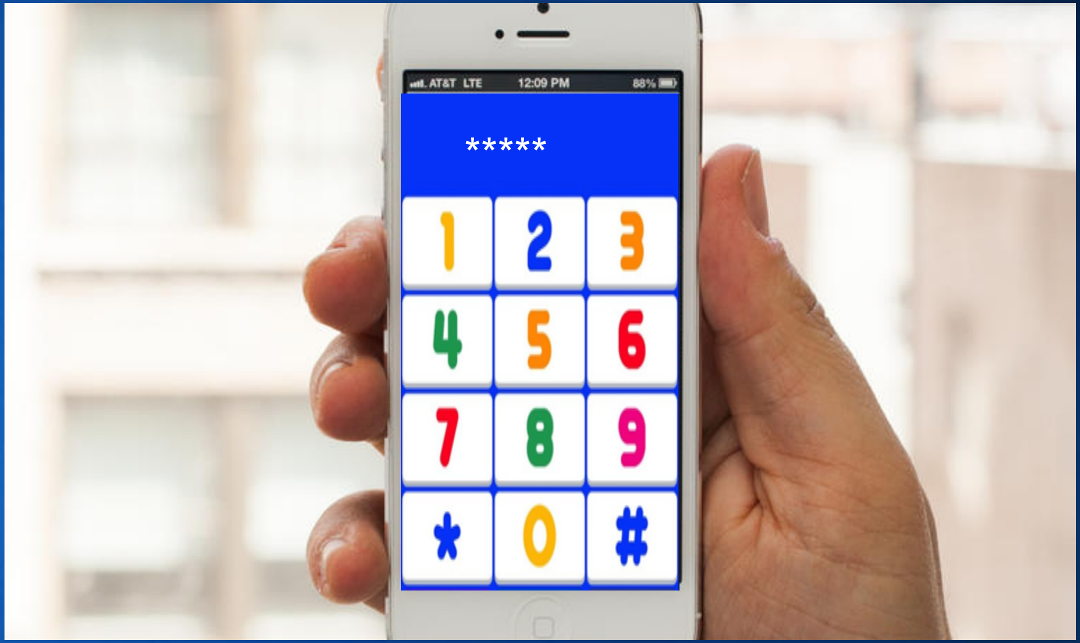


Call support

Explicit Textual-Input is **unavoidable**

Our idea for protection

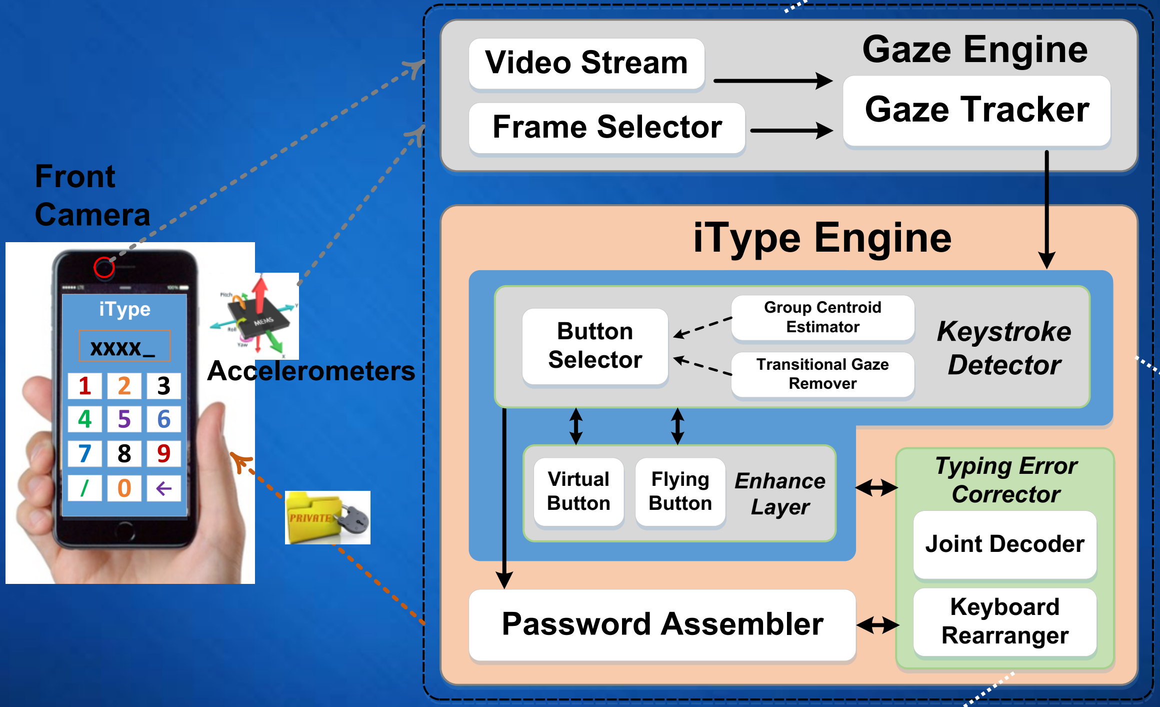
- Eye gaze for input
  - Front camera



- Secure
  - Back
    - A keyboard
  - Front
    - Difficult to distinguish
    - Keyboard layout may change

iType framework

3. Noises from device motions



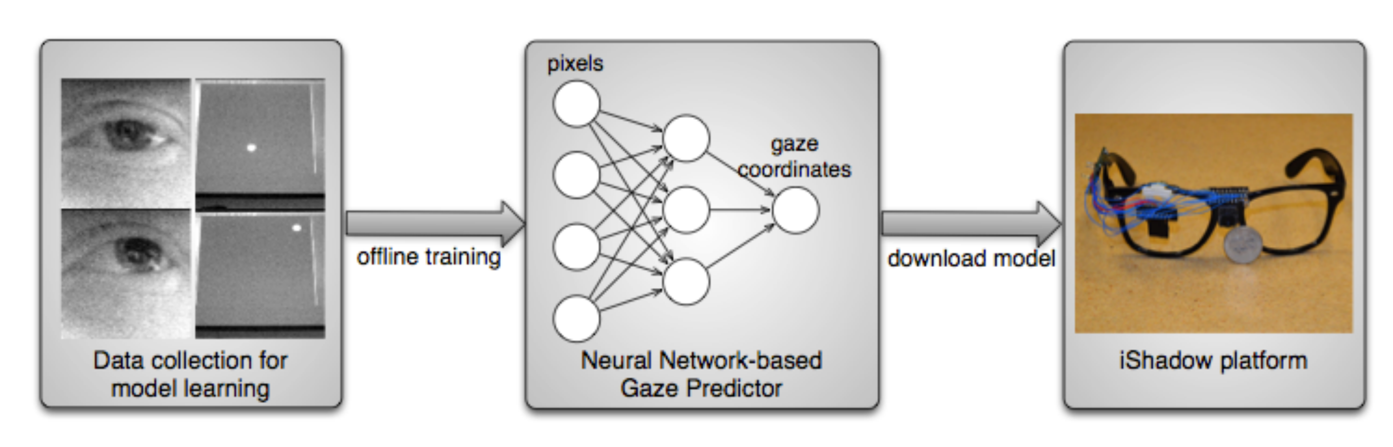
1. Unreliable mobile gaze tracking

2. Lack of true text-entry value in error correction

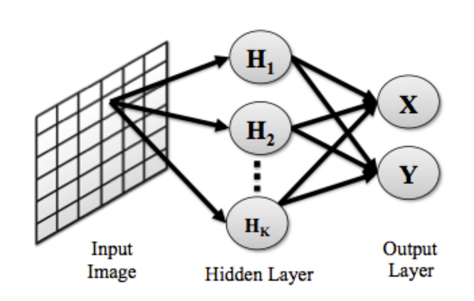
Unreliable mobile gaze tracking

- **Problem statement:**

Gaze tracker training [5]:


Data collection for model learning
offline training
pixels
gaze coordinates
Neural Network-based Gaze Predictor
download model
iShadow platform

Input: image



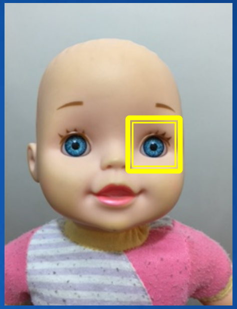
Output: gaze coordinates

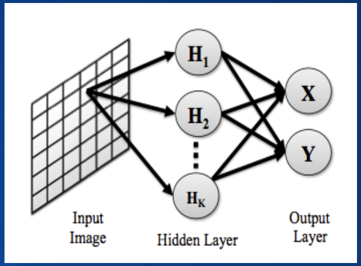
[5] **"ishadow: design of a wearable, real-time mobile gaze tracker", in Proc. of ACM MobiSys, 2014.**

Unreliable mobile gaze tracking

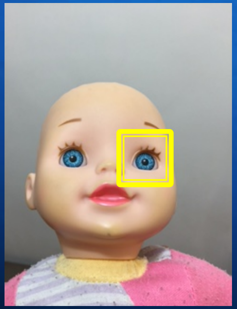
- **Problem statement:**

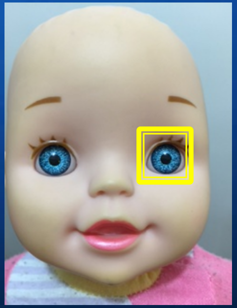
For mobile devices:





Training



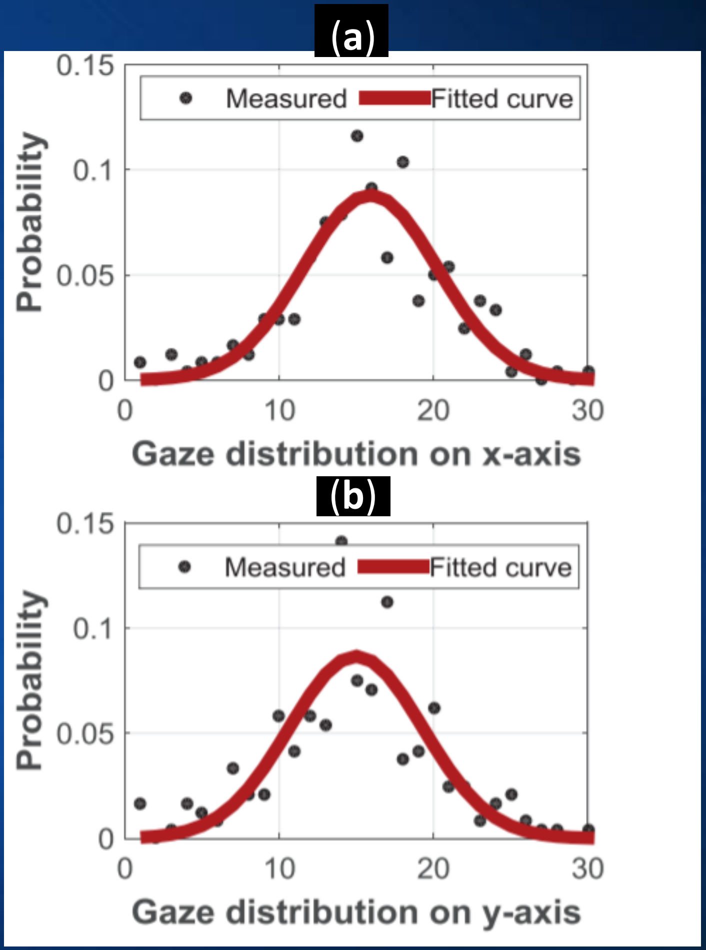


Unreliable mobile gaze tracking

- Observations

x-axis

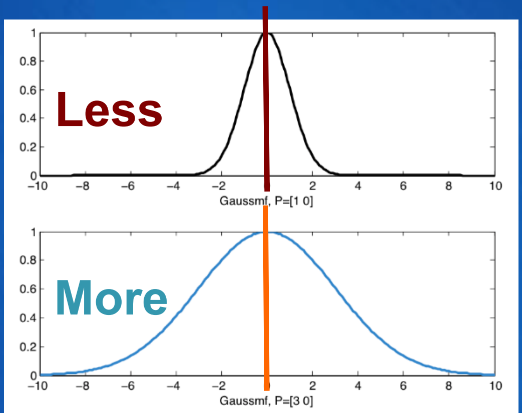
y-axis



Unreliable **tracking**

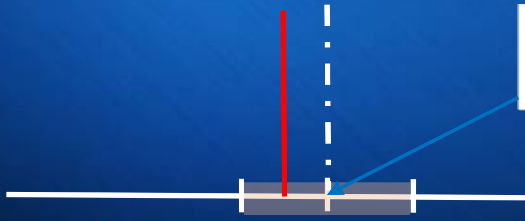
Unreliable mobile gaze tracking

- Formal description



Min. samples to achieve certain confidence?

- Solution overview (*n* gaze points)



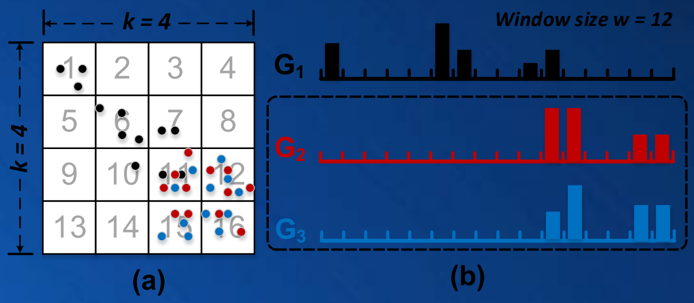
$$\left(\bar{x} - \frac{t_{\frac{\alpha}{2}}}{\sqrt{n}} S_x, \bar{x} + \frac{t_{\frac{\alpha}{2}}}{\sqrt{n}} S_x\right)$$

$$S_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2$$

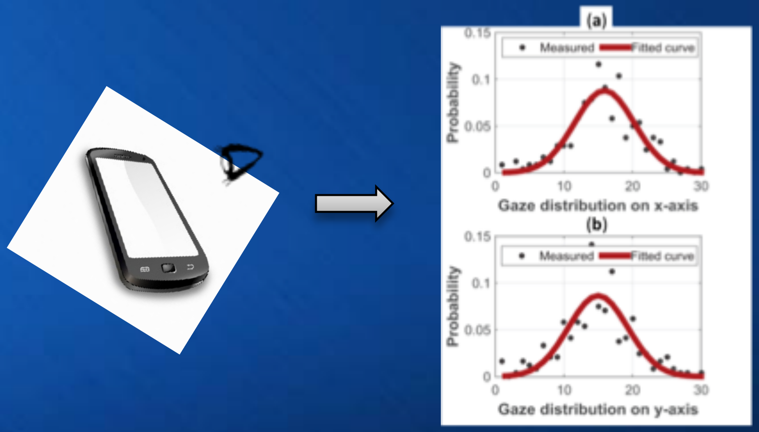
At least (1- alpha)

Keystroke detection

- When to start:
    - KL divergence

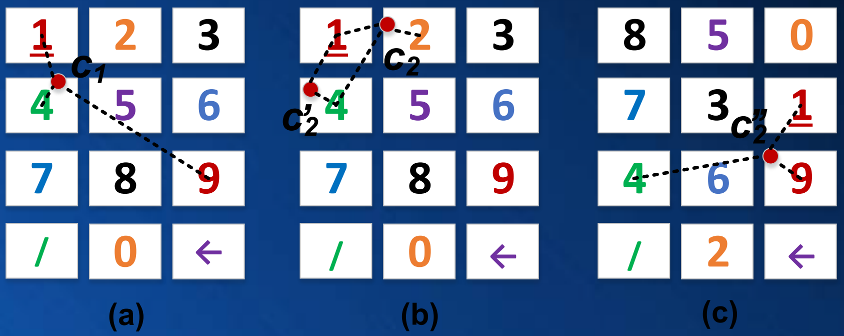


- When to stop:
    - Approximation

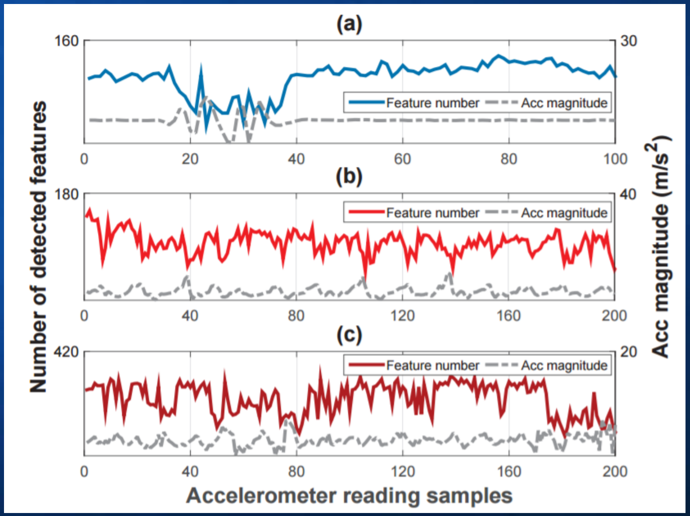


Other modules

- Input error correction
  - Joint decoding

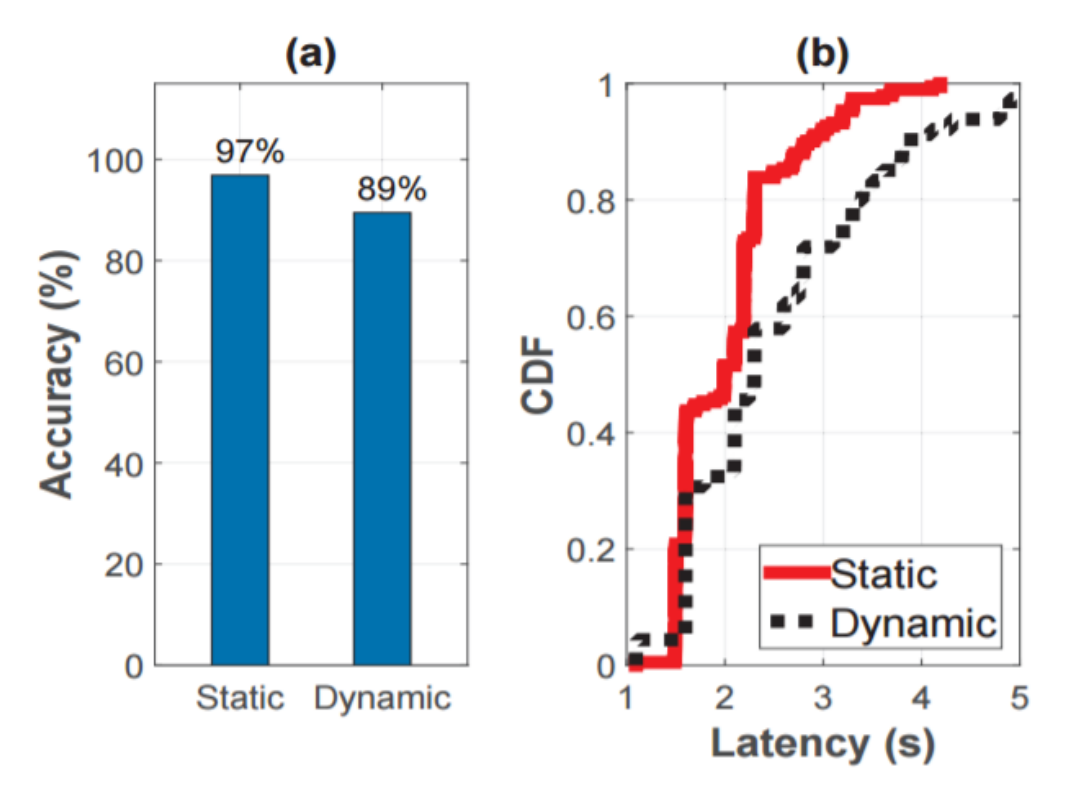


- Frame selection
  - Sensor-assisted



Evaluation

- Overall performance



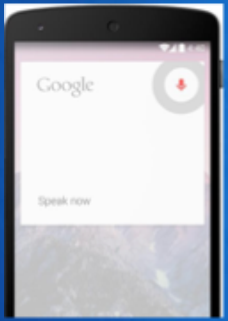
Individual keystroke:

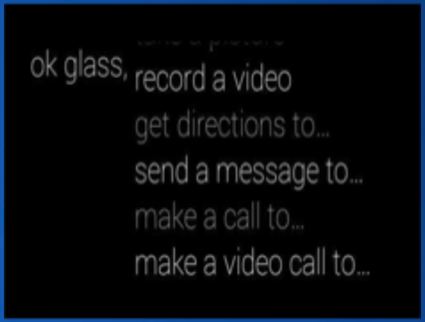
- **Accuracy**
  - Static: 97%
  - Dynamic: 89%
- **Latency**
  - Static: 2.0s
  - Dynamic: 2.6s

Unconventional Sensing

Voice Control

- Popular on smartphones
  - Electronic assistant (Google now, Apple Siri)
- Primary method of interaction for wearables (smartwatch, smartglass)
  - Touch not always feasible for wearables





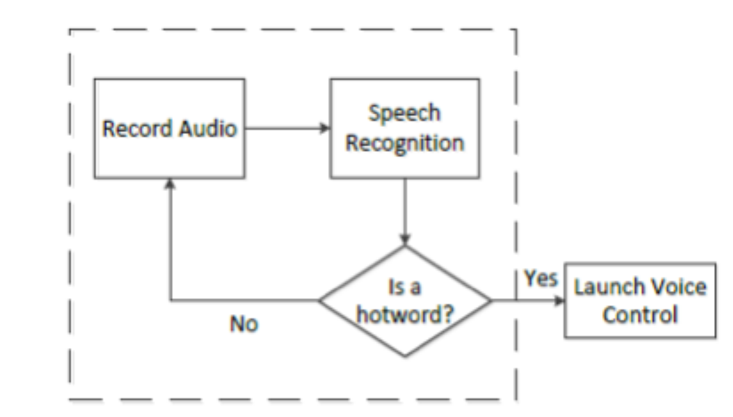
ok glass, record a video
get directions to...
send a message to...
make a call to...
make a video call to...

Will it rain tomorrow?
Play music by Bruno Mars
Add gelato to my shopping list
When is Thanksgiving?
What's the weather in Los Angeles this weekend?
amazon

- Internet-Of-Things (IoT) applications
  - Low-cost, low-power, pervasive
  - Example: Amazon Echo smart speaker

Current Voice Control Applications

- Hotword detection
  - Detect the hotword “Ok Google”, “Hi Galaxy” etc. to start voice control
  - Distinguishes between voice command and other conversations

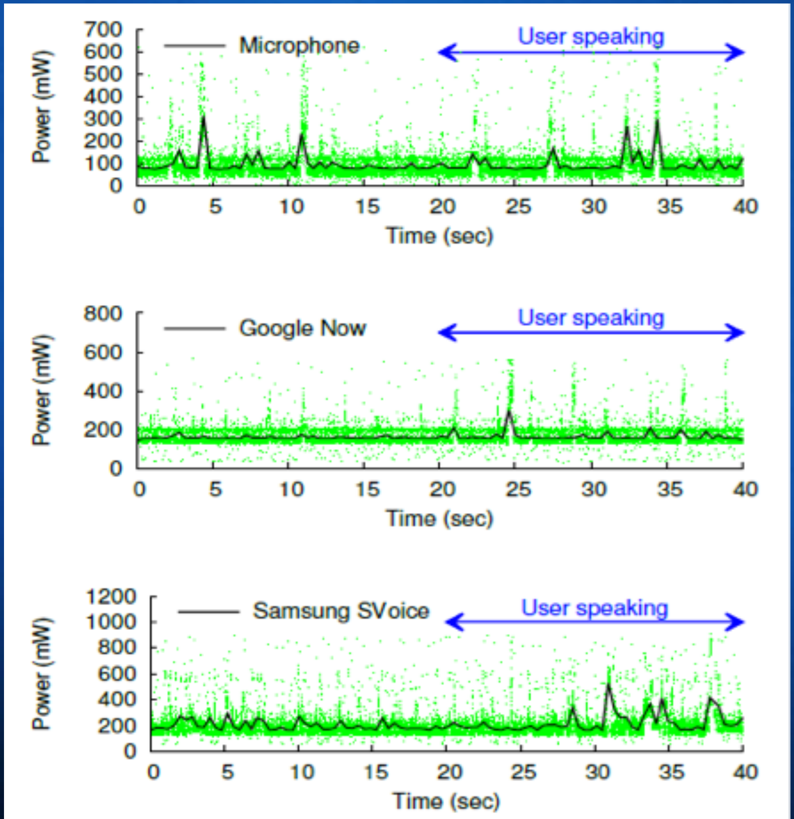


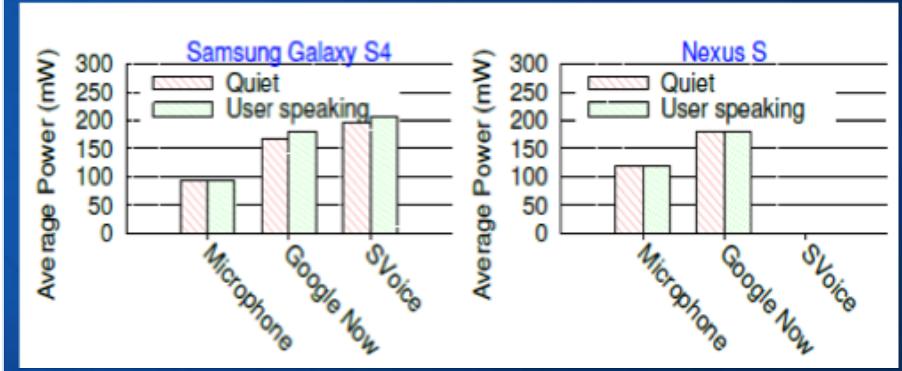
Hotword Detection

- Continuous audio sensing
  - “Always” listen for hotword
  - Energy expensive
  - Unsuitable for low-power devices

Motivation - Energy Hungry Voice Control

- Current voice control and hotword detection is energy inefficient
  - Microphone sampling rate - 44 KHz

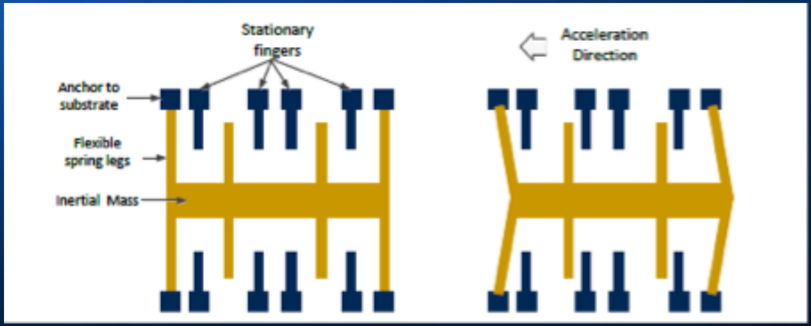




AccelWord - Hotword Sensing using Acclerometer

- Accelerometer sensor
  - Included in almost all smart devices (phones, watches, glasses etc.)
  - Primary purpose to sense motion
  - Low-cost (< $5) and low energy (sampling < 200 Hz)

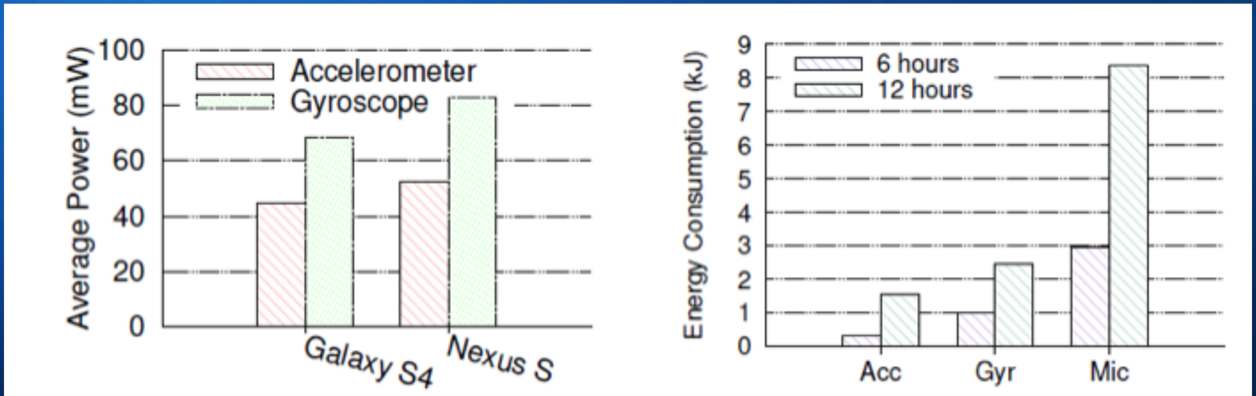
- AccelWord idea
  - Empirical evidence that accelerometers are sensitive to spoken voice
  - Accelerometer registers acceleration when audio signals strike the inertial mass
  - *Can we "listen" using accelerometer?*



Hotword Detection using Accelerometer

- Can we use accelerometer to “listen” for hotwords?
  - If the hotword is detected using accelerometer, start the microphone for complete voice recognition

- Advantage - lower energy consumption compared to microphone
  - 20 Hz - 22 KHz human voice modulated on 200 Hz accelerometer samples
  - Lower sampling results in low-power sensing



(c) The Average Energy Consumption of a 30 minutes Trace

(d) The Long Term Total Energy Consumption of Three Sensors

AccelWord Challenges

- Hotword recognition
  - Can accelerometer distinguish between hotword and other words?
  - Complete speech recognition is difficult

- Human mobility interference
  - How to remove the mobility-related acceleration to distill voice-related acceleration data?

- Noise cancellation
  - Advanced techniques already exist for microphone to remove background noise
  - Is the impact of background noise on accelerometer too detrimental?

- *Security Threat !*

Performance Evaluation

- 10 volunteers
  - 5 females and 5 males

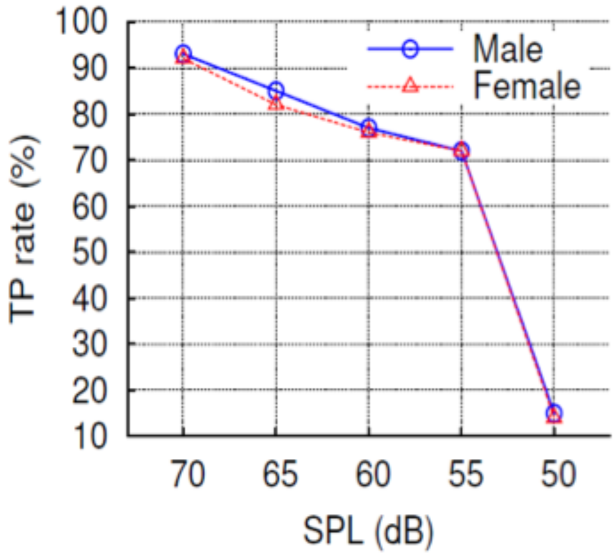
- Two smartphones
  - Samsung Galaxy S4 and Google Nexus S

- Comparison with
  - Google Now and Samsung S Voice
  - TP rate, FP rate and energy

- Training and testing instances
  - Hotword instances - 5 mobile, 5 stationary (100 times)
  - Other random sentences - 20 (200 times)

Hotword Detection Accuracy

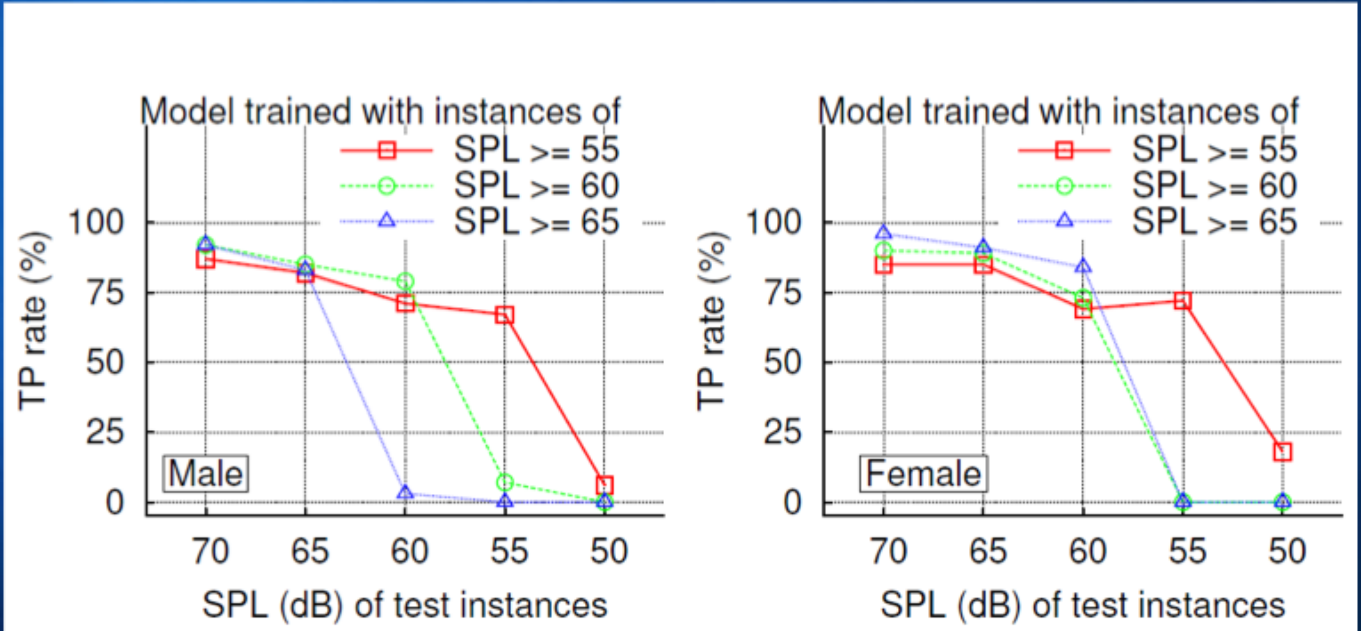
- Trained and tested with the same SPL
  - TP rate measures the fraction of hotword instances correctly detected from all spoken instances including random sentences



	FP Rate (%)	
SPL (dB)	Male	Female
70	4.8	4.1
65	5.5	5.1
60	7.8	7.5
55	8.5	8.0
50	1.8	2.1

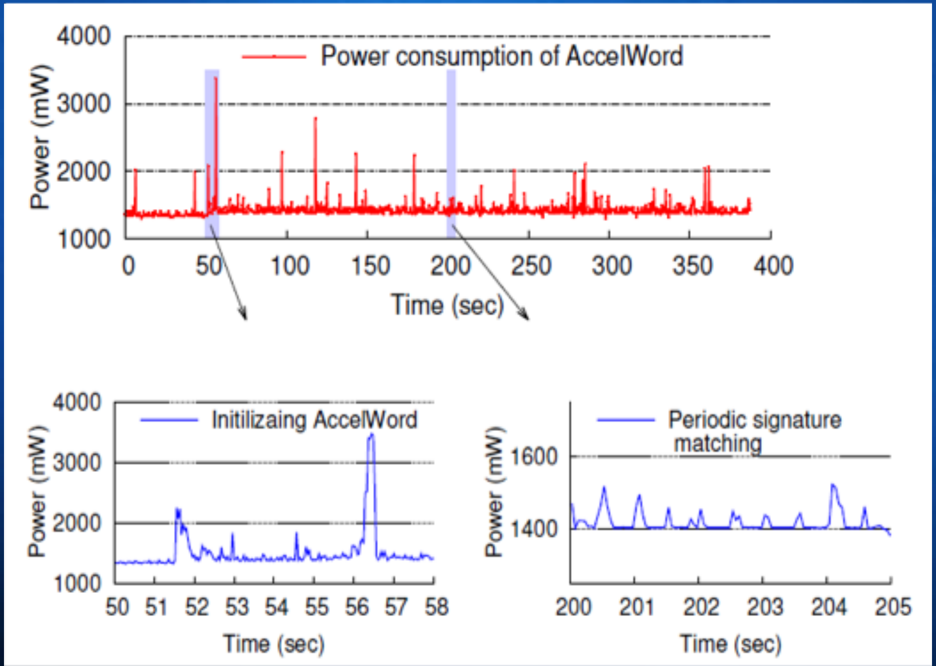
Hotword Detection Accuracy

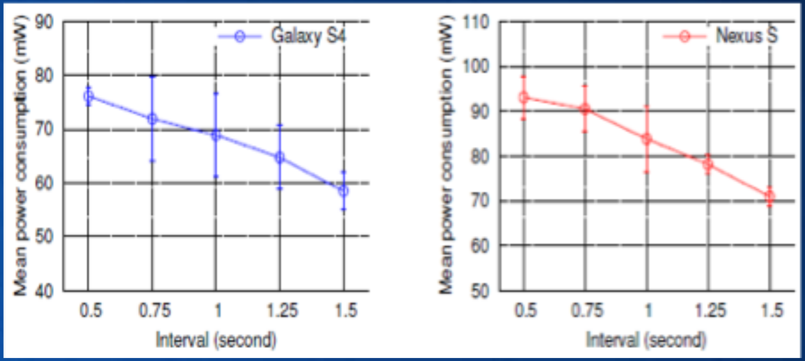
- Trained and tested with different SPL
  - Lower TP rate compared to the case where classifier is trained and tested with same SPL

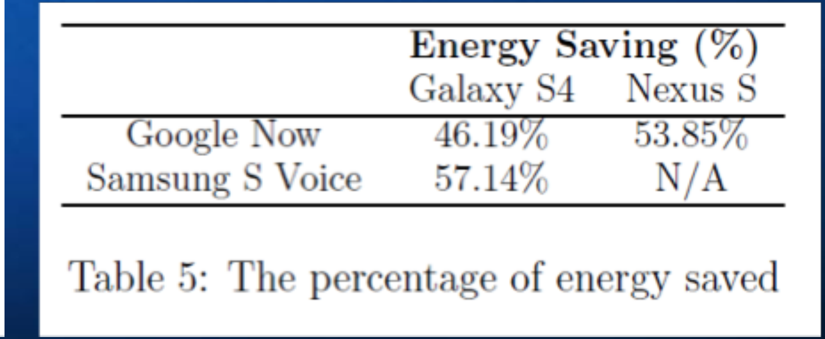


Energy Efficiency

- Energy savings mostly attributed to low-cost sensing through accelerometer
  - With optimized implementation of AccelWord, further processing-related savings can be achieved







	Energy Saving (%)	
	Galaxy S4	Nexus S
Google Now	46.19%	53.85%
Samsung S Voice	57.14%	N/A

Table 5: The percentage of energy saved

Securing the IoTs of Future!

- *More and more IoTs will be invading the space around us*
- *Exploitation of cyber sensing and physical sensing can be done in an integrated manner*
- *IoTs will learn and adapt to the environments*
- *Adversarial IoTs will evolve*
- *Safeguarding Adversarial Machine Learning will bring in complex challenges*
- *Containment and isolation of compromised IoTs will be a new topic of research*

TY