SCIENCE
PASSION
TECHNOLOGY

AndroGUARD: Mitigation of Sensor Fingerprinting on Android

Gergö Kranz

20.02.2025

Outline

Introduction

- Misuse of the Android API
- Used for targeted advertisements
- Does not require user permission

Introduction

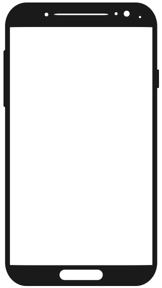
- Misuse of the Android API
- Used for targeted advertisements
- Does not require user permission

Introduction

- Misuse of the Android API
- Used for targeted advertisements
- Does not require user permission

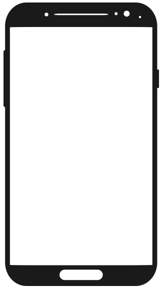
Smartphone Fingerprinting

- Similar to browser fingerprinting
- Not as known as browser fingerprinting
- Zero permission identifiers
- Personalized configurations



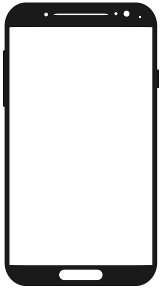
Smartphone Fingerprinting

- Similar to browser fingerprinting
- Not as known as browser fingerprinting
- Zero permission identifiers
- Personalized configurations



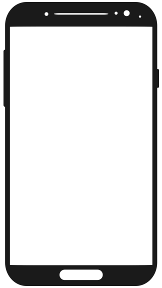
Smartphone Fingerprinting

- Similar to browser fingerprinting
- Not as known as browser fingerprinting
- Zero permission identifiers
- Personalized configurations



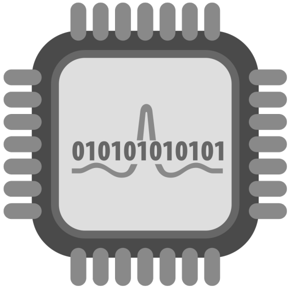
Smartphone Fingerprinting

- Similar to browser fingerprinting
- Not as known as browser fingerprinting
- Zero permission identifiers
- Personalized configurations



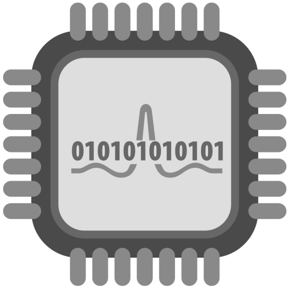
Fingerprinting Sensors

- Measurement inaccuracy of sensors
- Simple to fingerprint via machine learning algorithmus
- Constant over the sensors lifetime



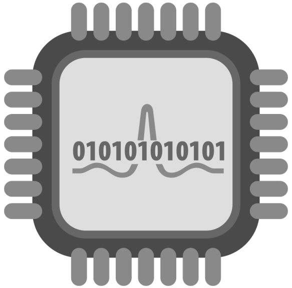
Fingerprinting Sensors

- Measurement inaccuracy of sensors
- Simple to fingerprint via machine learning algorithmus
- Constant over the sensors lifetime



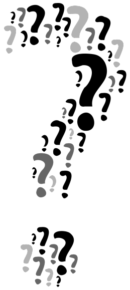
Fingerprinting Sensors

- Measurement inaccuracy of sensors
- Simple to fingerprint via machine learning algorithmus
- Constant over the sensors lifetime



Main Question

How to protect against sensor fingerprinting



Proposed Solutions

Calibration

- Systematic adjustment of sensor readings
- Correcting the sensor data

Noise Generation

- Introduces variability into the sensor data
- Masks the original values

Challenges

Calibration

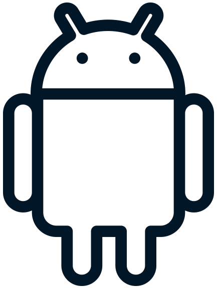
- Requires user awareness and interaction
- Requires precision

Noise Generation

- Degrade the functionality of applications
- Code has to be modified

Our Methodology

- Noise Generation
- Patch application vie A2P2 framework



Modifying the Sensor API

- Intercept calls to registerListener method
- Provide modified values to onSensorChanged method

Noise Generation

- Adds random gain and offset to every value
- Masks values
- Loss of precision

Implementation

- Intercept Method
- Noise Generating Function
- Random Value Generation Function

Intercept Method

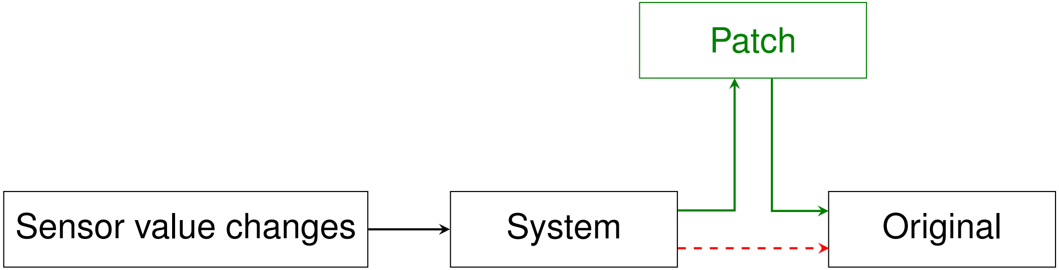


Figure: The function calls from the system are intercepted by our patch and forwarded after modification to the original function.

Implementation

- Intercept Method
- Noise Generating Function
- Random Value Generation Function

Noise Generating Function

$$value_{new} = \frac{(value_{old} - offset_{sensor})}{gain_{sensor}}$$

Implementation

- Intercept Method
- Noise Generating Function
- Random Value Generation Function

Application of Patch

- Straightforward application
- Only requirements are
  - JAVA JRE
  - A2P2
  - APK to be modified

Testing

- Functionality
- Effectiveness
- Usabilty

Functionality

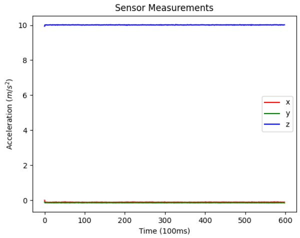


Figure: recorded values before the patch

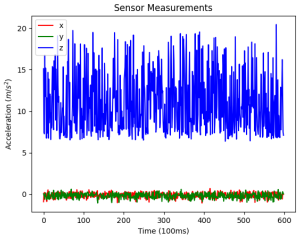


Figure: recorded values after the patch

Testing

- Functionality
- Effectiveness
- Usabilty

Effectiveness

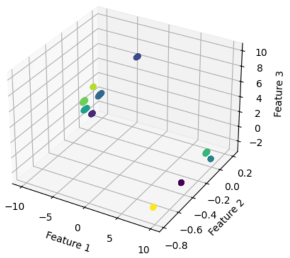


Figure: knn decision boundaries before the patch

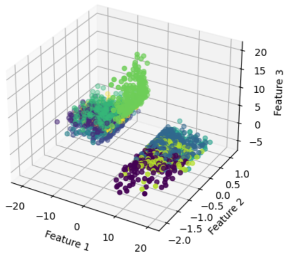


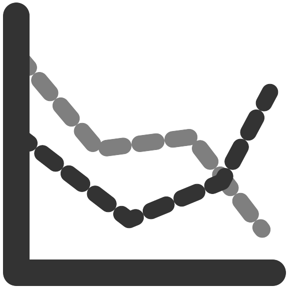
Figure: knn decision boundaries after the patch

Testing

- Functionality
- Effectiveness
- Usabilty

Noise Level Adjustment

- Increasing noise decreases fingerprintability
- Increasing noise decreases functionality



Discussion & Limitations

- Comparing values before and after the patch
- Could not be done sufficiently due to limited access to supported hardware

Conclusion

- Masking the sensor values decreases fingerprintability
- Modifying the SensorEventListener makes it easy to incorporate the patch into the Android API