ANDROID PIN CRACKING

Presented by:

Nidheesh Panchal (2016UCP1008)

Siddhant Gupta (2016UCP1455)

Rahul Jangir (2016UCP1396)

Guided by:

Dr. Vijay Laxmi

Abstract

- Numeric PINs of a smartphone to be predicted by an attacker using motion data.
- Zero-permission motion sensors play crucial role in exploiting this vulnerability.
- No special permissions required for accessing motion sensors.
- Website opened in a browser or an application in background can collect data.
- The pattern of each key press is observed and a suitable model is trained to identify individual digits.

Why Android?

- Two major smartphone operating systems are Android and iOS.
- Market share of Android and iOS is 81.7% and 17.9%, respectively.
- Targeting Android devices means covering larger portion of all smartphones.

Android Sensors

- Motion Sensors
  - *Position Sensor*
  - *Accelerometer Sensor*
  - *Gyroscope Sensor*
  - *Gravity Sensor*
- Environmental Sensors
- Position Sensors

Zero-permission Sensors

- Range of zero-permission sensors are found in modern smartphones to enhance user experience. e.g. – Proximity, Gyroscope, Accelerometer, etc.
- These pool of sensors may unintentionally leak sensitive information.
- These sensors are accessible without user permissions, so an attacker can take advantage of an unaware user.
- Sensors can reveal privacy-related information about the user, such as personal identification number (PIN) or movement patterns.

Sensors Used and Why?

- Type of sensors used are Motion Sensors (Accelerometer, Gyroscope and Gravity)
- User enters PIN on numerical keypad where each possible number is in the range of (0-9).
- While entering the numbers, user moves fingers to reach the number.
- Smartphone will tilt, rotate or move.
- These movements are minute and could be easily captured by motion sensors.

Accelerometer Sensor

Description:

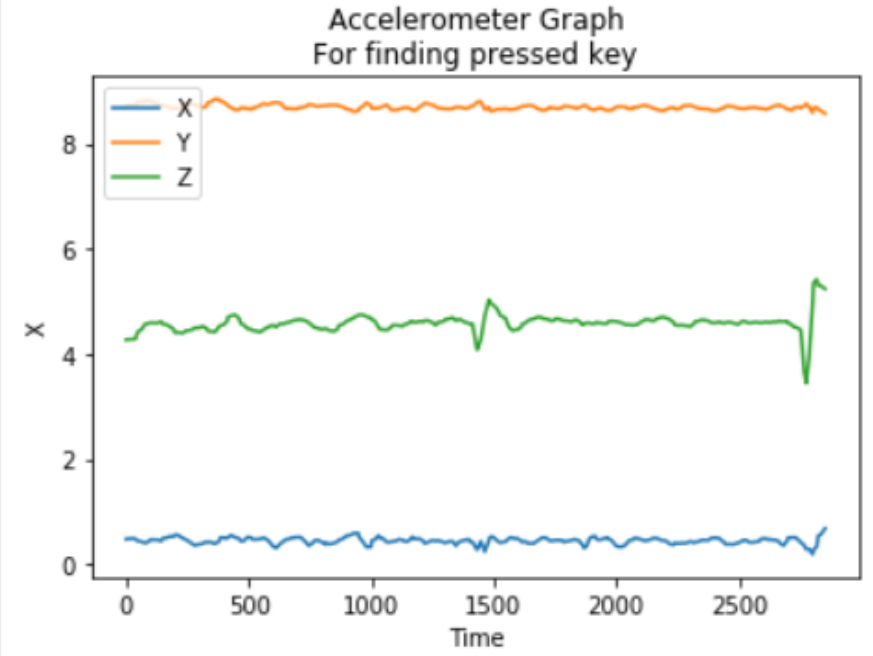
Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), may or may not include the force of gravity.

Uses:

Linear motion detection (shake, tilt)

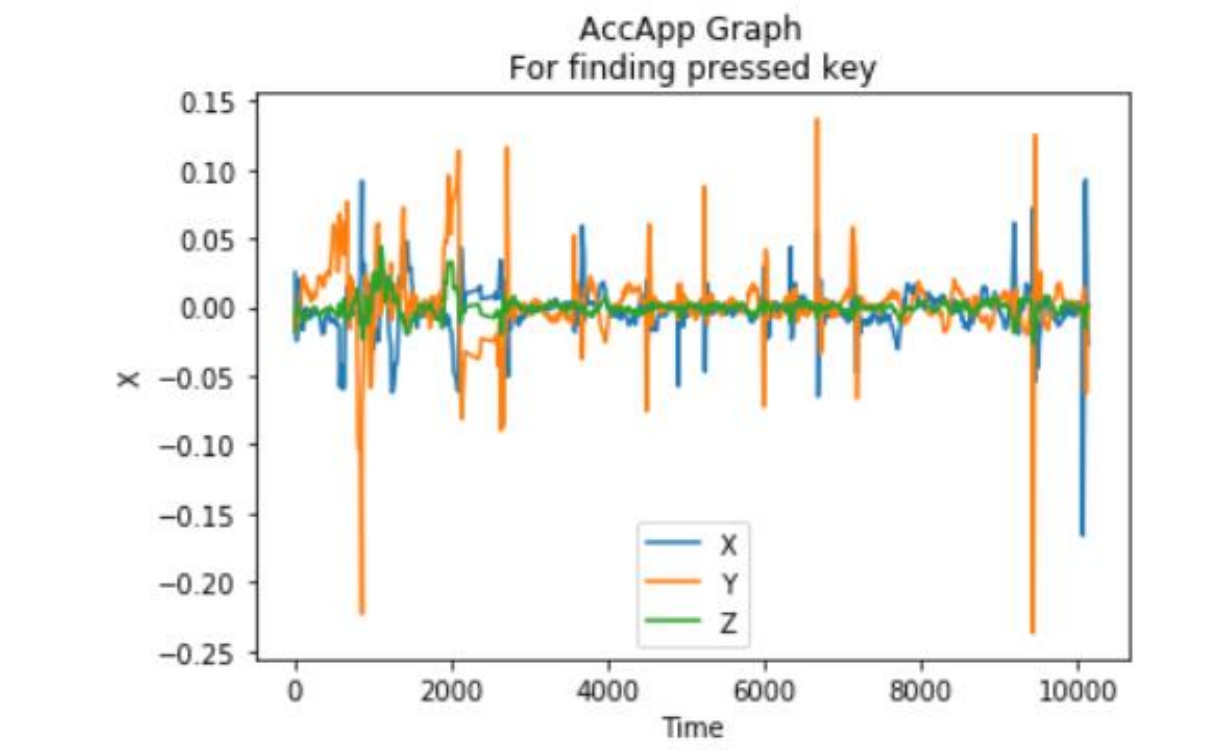
	Time	X	Y	Z
0	0	0.476446	8.705317	4.268862
1	7	0.483629	8.726865	4.273650
2	35	0.488417	8.729259	4.280833
3	38	0.481235	8.741230	4.283227
4	48	0.442927	8.791509	4.422090

Accelerometer



Accelerometer Sensor

We can see different patterns when different keys are pressed.



Gyroscope Sensor

Description:

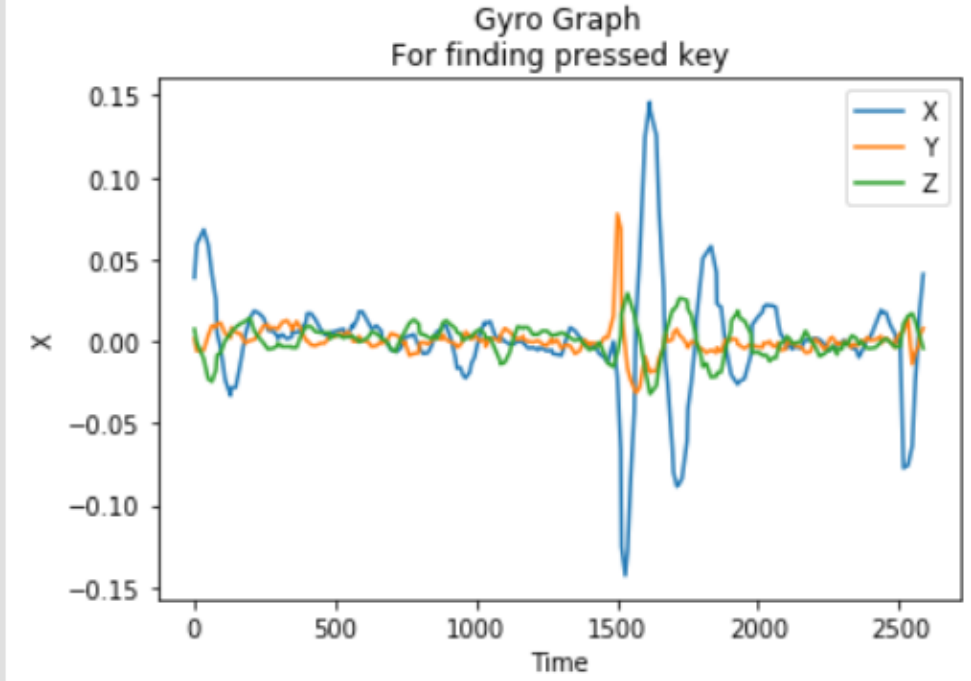
Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).

Uses:

Rotation detection (spin, turn).

	Time	X	Y	Z
0	2	0.038882	0.001598	0.007457
1	10	0.059122	-0.005859	-0.000533
2	35	0.068177	-0.005326	-0.010653
3	51	0.058590	0.003728	-0.022903
4	64	0.041013	0.009587	-0.024501

Gyro



Gravitation Sensor

Description:

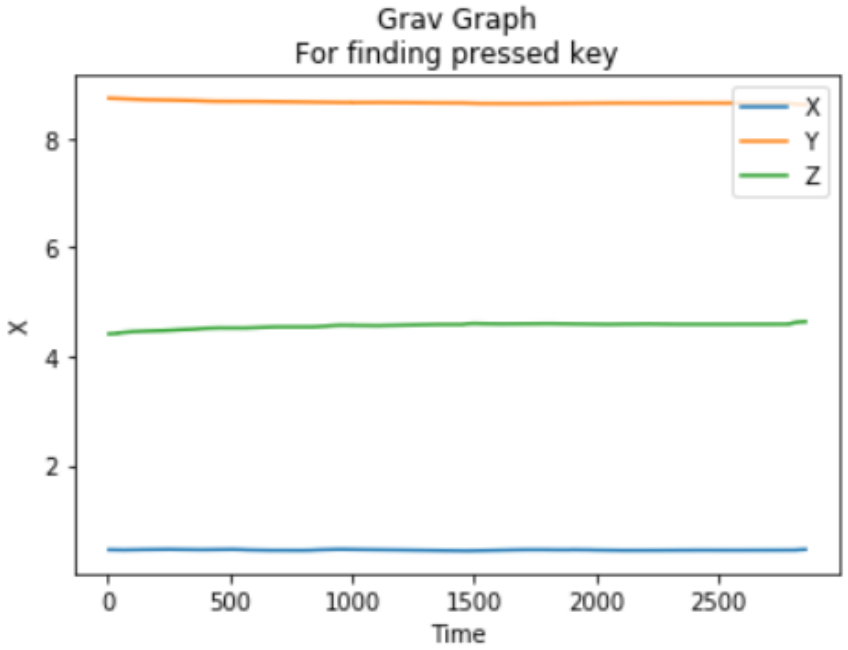
Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).

Uses:

Motion detection (shake, tilt).

	Time	X	Y	Z
0	5	0.448015	8.745902	4.413486
1	8	0.447603	8.745322	4.414678
2	14	0.447336	8.744362	4.416605
3	37	0.447088	8.743085	4.419162
4	40	0.446618	8.741626	4.422093

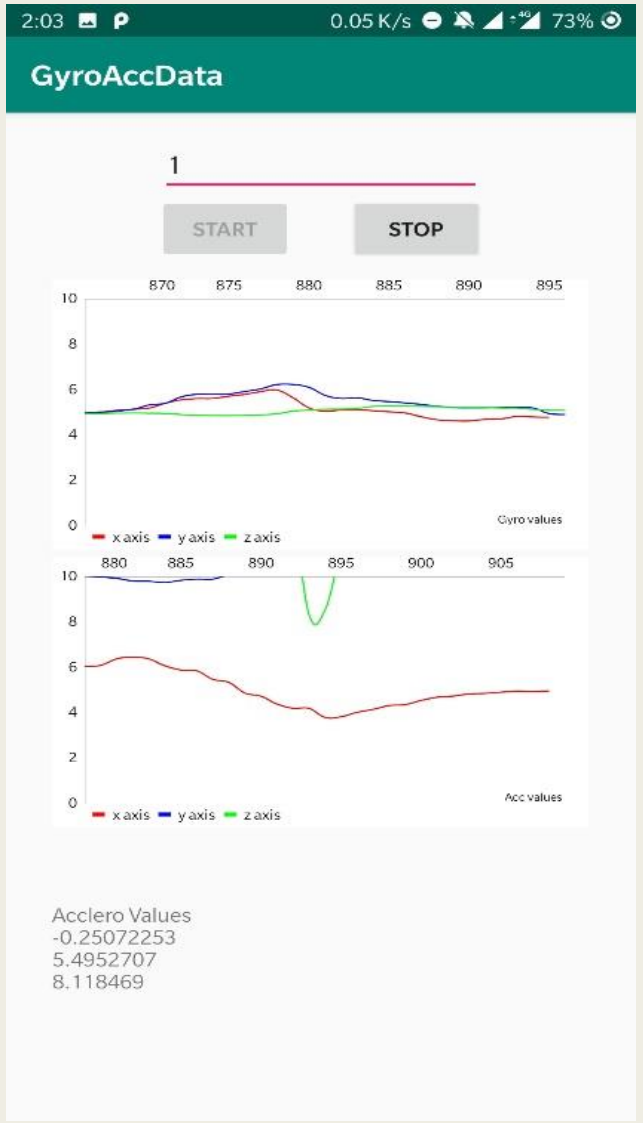
Grav



Behavior of Sensors

- Sudden spike is seen in the graph when the screen is touched.
- The top graph shows the gyroscope and bottom graph shows accelerometer sensor data plotted against time.

Red: x-coordinate
Blue: y-coordinate
Green: z-coordinate

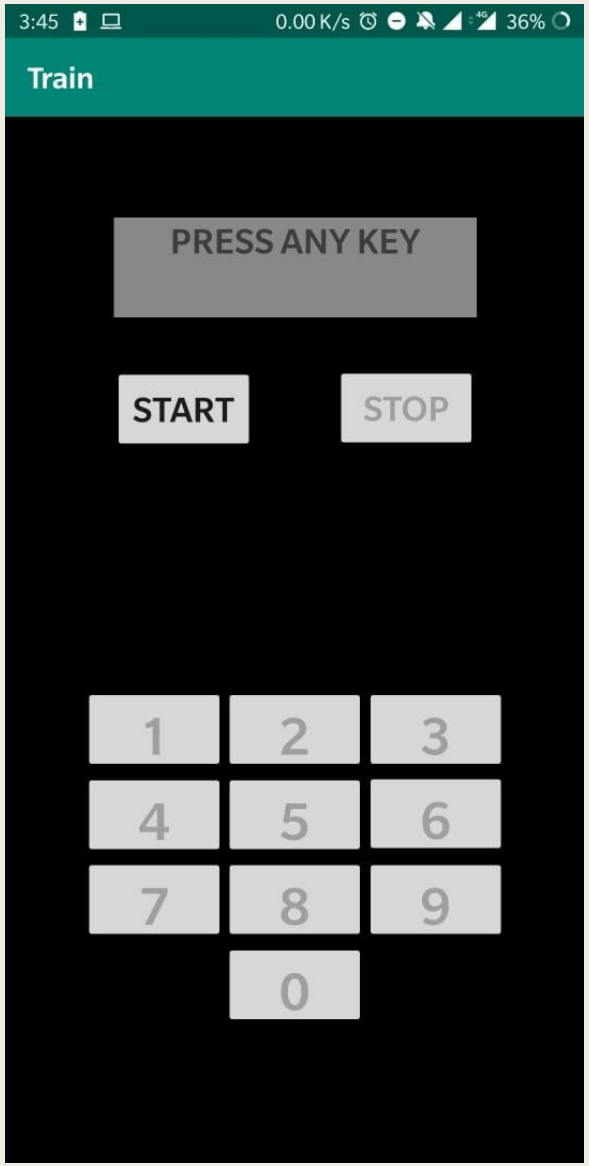


Proposed Method

- After observing the results from the experiments performed, we propose a method to identify individual digits of a PIN.

Android Application

- An Android application was built to collect motion sensor data for model training.
- Application named as **Train**.
- Android application was chosen over web browser because sampling rates available in browsers are much lower than those in mobile application.



Data collection

- Starting with a basic and simple data collection from a single user. Using the android application “Train”.
- A user is made to either sit or stand and hold the phone in right hand in one identical position every time.
- The app collects 4 types of sensor data
  - *Accelerometer with gravity*
  - *Accelerometer without gravity*
  - *Gyroscope*
  - *Gravity*

Preprocessing

- The raw data collected from the application contains multiple rows of X, Y, Z coordinate values and time.
- Collected the timestamp of *onKeyDown* and *onKeyUp actions* and the number key that was pressed.
- Three ways of data processing were used and compared the accuracy with different classification model.
  - Windowing
  - Three values
  - Three values from uniform data.

Classification model

- Following models are used for classification.
  - *Random Forest*
  - *Decision Tree*
  - *Logistic Regression*
  - *Naïve Bayes*
  - *KNN*
  - *SVM*
  - *Multioutput*
- Taking all types of pre-processed dataset and taking that as input to all models running in same configuration for each(to do justice to the comparison), comparison is done.

Using the sklearn library for python, the listed classifiers are implemented and the results are compared.

Experimental Results

- It was observed that motion detected by the gravity sensor did not show remarkable changes in sensor data values, hence did not contribute to give accurate prediction. (16%).
- The sensor data from the sensor type accelerometer without gravity also did not give accurate results (12.3%).
- The following table shows maximum accuracy among all types of processing methods.

Type of data	Random Forest		Decision Tree		Logistic		Naïve Bayes		KNN		SVM	
	Acc	Gyro	Acc	Gyro	Acc	Gyro	Acc	Gyro	Acc	Gyro	Acc	Gyro
On-table	37.5	42.5	20	34	19	19	22.5	23.75	35	28.75	17.5	35
In-hand	38.93	40.7	27.65	31.2	36.87	41.13	26.54	33.62	32.74	42.47	33.62	42.47
Hybrid	34.71	31.6	29.87	29.46	19.91	26.14	12.43	22.27	36.78	39.89	26.94	27.46

Multioutput Classification Results

- In multioutput classification, 4 outputs are obtained and the accuracy is calculated as: number of times the actual key pressed label is present in this list of 4 outputs divided by size of test data. “List of 4” column in the following table, displays this accuracy measurement and “First 2” column displays accuracy of the actual key press present in the first 2 position from the list of 4.
- The following table shows maximum accuracy among all types of processing methods.

	Multioutput			
Type of data	Acc (%)		Gyro (%)	
	First 2	List of 4	First 2	List of 4
On-table	41	62	44	61
In-hand	49.64	72.34	59.57	73.75
Hybrid	43.56	61.82	51.45	67.63

Conclusion

- Exploring the sensors used on smartphones and exploiting the vulnerability, it is possible to develop a single-digit classification methodology to recover PIN from maliciously captured sensor data.
- Using different types of preprocessing methods and different classification models to classify a single digit, the maximum accuracy achieved for on-hand data collection method is 42.5% with random forest, in-hand data collection method is 42.47% with SVM and merged data is 39.89% with KNN.
- The next steps include, finding new preprocessing method, classification model and then selecting the best out of the lot. Training the model with the collected data and creating either an Android background app service to exploit sensor data or host a website.

THANK YOU