smartphone Side-Channel Attacks and Defenses

**Module 2 lAB Manual**

**Lab Manual Development Institution:** Colorado School of Mines

**Lab Manual Contributors**:

Zhiju Yang\*, Ryan Jackson+, Chuan Yue@

(\*: PhD student and research/teaching assistant; +: Master Student; @: Faculty.)

# Motion sensor data preprocessing

**Lab Description:** In this lab, you will construct tools to preprocess smartphone motion sensor data. You will design algorithms and build tools to segment motion sensor data (of multiple consecutive keystrokes collected in a certain time window) into individual data pieces each corresponding to an individual keystroke (e.g., typing digit ‘8’ on the soft-keyboard of the smartphone), and evaluate the segmentation correctness and accuracy. In more details, you will use a dataset consisting of the motion sensor data collected from 30 users related to their typing of PINs, and will segment the motion sensor data of each PIN typing for each user. The segmented data will be used in the follow up labs for performing inference attacks. We suggest you to use Python to write your tools so that the preprocessed data can be easily used in the follow up labs. This lab consists of five STEPs.

The high-level **learning outcomes** and the corresponding **assessment** of this lab are summarized as follows. In other words, upon completion of this lab, students should be able to:

* **Propose** ideas for preprocessing or segmenting motion sensor data.
  + Assessed by the tasks and outputs specified in STEP 1.
* **Construct** a tool to separate the motion sensor data for the typing of each 4-digit PIN.
  + Assessed by the tasks and outputs specified in STEP 2.
* **Construct** a tool to inspect the correctness of the typed PINs and discard incorrect data.
  + Assessed by the tasks and outputs specified in STEP 3.
* **Construct** a tool to segment the motion sensor data of each keystroke.
  + Assessed by the tasks and outputs specified in STEP 4.
* **Construct** a tool to save motion sensor data of different types of keystrokes for all the users.
  + Assessed by the tasks and outputs specified in STEP 5.

**Lab Environment:** Linux, Mac, or Windows.

**Lab Files that are Needed:** TheLab Manual file and the CSM\_MotionSensor\_Dataset.zip file.

**Learning Setting:** This lab module is for students to complete outside the classroom, so it can be used in either face to face or online courses.

**Prerequisites:** Java or Python Programming, Basic Cybersecurity and Machine Learning knowledge and skills, Linux or Windows Systems, Computer Networks.

**Length of Completion:** 600 minutes.

**Level of Instruction:** Senior undergraduate students or graduate students in CS or related STEM programs. The lab exercise should be further simplified if it will be used for freshmen, sophomores, or none-CS major students.

**Interconnection with Other Labs:** This lab module is standalone by itself; however, if needed, an instructor can use the details in the course project manual and the other four lab manuals to provide additional hints to students.

**Assessment Guideline:** Students should follow the steps to answer all the questions. Based on the points assigned to each individual question, the instructor will grade each answer (together with the additional materials if specified for the question) in terms of its correctness (60%), clarity (20%), and concision (20%).

### **Lab Exercise/step 1 (download the dataset and learn about it)**

Download and unzip the file CSM\_MotionSensor\_Dataset.zip, which is a dataset of the motion sensor data collected from 30 users related to their typing of PINs. Each user typed 50 different 4-digit PINs for 5 times, so the motion sensor data collected from each user correspond to the typing of 250 PINs with 1,000 digits in total and are saved in a specific file for each user. We consider the motion sensor data corresponding to each key (or keyboard) typing as the data incurred after the current key is pressed and before the next key is pressed.

Each file for a user has the same format as described in Table 1 with some details described in Table 2. In Table 1, the contents enclosed in the double quotation marks are fixed (i.e., hard-coded), and they will be used for you to segment the motion sensor data.

Note that to simplify the lab, we adopted (with minor changes) this simple file format shown in Tables 1 and 2 from the research paper “Stealing PINs via Mobile Sensors: Actual Risk versus User Perception” (By Maryam Mehrnezhad, Ehsan Toreini, Siamak F. Shahandashti, Feng Hao. On arxiv.org with ID 1605.05549, 2017). We used a more comprehensive file format in our research papers “Cross-site Input Inference Attacks on Mobile Web Users” (By Rui Zhao, Chuan Yue, and Qi Han. In proceedings of the International Conference on Security and Privacy in Communication Networks 2017) and “Effective Mobile Web User Fingerprinting via Motion Sensors” (By Zhiju Yang, Rui Zhao, and Chuan Yue. In submission and to appear in a conference, 2018), but that format is too complex for a lab exercise.

Table 1 The format of a motion sensor data file.

|  |  |
| --- | --- |
| **Line #** | **Line Content** |
| 1 | User ID |
| 2 | “User Starts” |
| 3 | “Typing Begins” |
| 4 | The first PIN displayed on your webpage (e.g., 5113) |
| 5, 6 | “Key Down”, “Key Up” (when the first digit is typed) |
| 7 to m | A series of motion sensor data for the typing of the first digit (as detailed in Table 2) |
| m + 1, m + 2 | “Key Down”, “Key Up” (when the second digit is typed) |
| m + 3 to n | A series of motion sensor data for the second digit |
| n + 1, n + 2 | “Key Down”, “Key Up” (when the third digit is typed) |
| n + 3 to p | A series of motion sensor data for the third digit |
| p + 1, p + 2 | “Key Down”, “Key Up” (when the fourth digit is typed) |
| p + 3 to q | A series of motion sensor data for the fourth digit |
| q + 1, q + 2 | “Key Down”, “Key Up” (consider as the end mark of the typing for this task) |
| q + 3 | User typed PIN, which may be different from the displayed PIN due to typing error |
| q + 4 | “Typing Ends” |
| q + 5 to t | The similar content with the same format as shown from Line 3 to Line q + 4 for the other 249 PINs |
| t + 1 | “User Finishes” |

Each series of motion sensor data are composed of a sequence of 9-tuples. Each 9-tuple is recorded at the same time consisting of the acceleration forces in the x, y, and z directions, the rotation rates in the alpha, beta, and gamma directions, and the acceleration forces with gravity in the x, y, and z directions. A series of motion data (e.g., content from Line 7 to Line m in Table 1) saved to a data file have the same format as shown in Table 2.

Table 2 The format of a series of motion sensor data.

|  |  |
| --- | --- |
| **Line #** | **Line Content** |
| 7 | accX\_1 (i.e., the first acceleration force in the x direction) |
| 8 | accY\_1 (i.e., the first acceleration force in the y direction) |
| 9 | accZ\_1 (i.e., the first acceleration force in the z direction) |
| 10 | rAlpha\_1 (i.e., the first rotation force in the alpha direction) |
| 11 | rBeta\_1 (i.e. the first rotation rate in the beta direction) |
| 12 | rGamma\_1 (i.e., the first rotation rate in the gamma direction) |
| 13 | accgX\_1 (i.e., the first acceleration force with gravity in the x direction) |
| 14 | accgY\_1 (i.e., the first acceleration force with gravity in the y direction) |
| 15 | accgZ\_1 (i.e., the first acceleration force with gravity in the z direction) |
| 16 | accX\_2 (i.e., the second ……) |
| 17 | accY\_2 (i.e., the second ……) |
| 18 | accZ\_2 (i.e., the second ……) |
| 19 | rAlpha\_2 (i.e., the second ……) |
| …… | …… |
| m | accgZ\_s (i.e., the s ……) |

**Question 1**: Please describe and explain your tentative idea on preprocessing (i.e., segmenting) the data contained in the provided dataset file.  
(Total score: 10 points. Grading rubric:  
100% points for a clear description and a clear explanation;  
60% points for only a clear description or a clear explanation;  
30% points for a vague description and/or a vague explanation.)

### **LAB EXERCISE/STEP 2 (Design and build a tool to separate the motion sensor data for each PIN)**

Data preprocessing is often a prerequisite for feature extraction and model training and testing in machine learning. Based on your tentative idea in step 1, you now will design and build a tool to separate the motion sensor data for each PIN. Your tool will read a single motion file for a user (e.g., User1Motion.txt) and separate the motion sensor data for each individual PIN. The result should be a list of 250 elements with each element containing the motion sensor data of one typed PIN. More accurately, each element should include the content from Line 4 to Line q + 3 as shown in Table 1, and should also include the User ID. For example, the format of the first element will be:

[Line 4 of Table 1, Line 5 of Table 1, ……, Line q + 3 of Table 1, User ID]

The format of the complete list will be:

(The 1st element, The 2nd element, ……, The 250th element)

Your tool may only need to keep this list in the memory and does not need to write it into a file because further processing will be performed in the next step.

**Question 2**: Please describe the design of your tool, and explain the data structures used in your tool.  
(Total score: 20 points. Grading rubric:  
100% points for a clear description and a clear explanation;  
60% points for only a clear description or a clear explanation;  
30% points for a vague description and/or a vague explanation.)

### **LAB EXERCISE/STEP 3 (Inspect the correctness of the PINs and discard the incorrectly typed ones)**

The dataset contains the typing of 250 PINs from each user, but some PINs could be incorrectly typed, i.e., the PIN typed by a user is different from the one displayed on the motion sensor data collection webpage. More specifically, Line 4 in Table 1 contains the displayed PIN, while Line q + 3 in Table 1 contains the PIN typed by a user. Improve your tool in Step 2 or write another tool to check the correctness of all the PINs, and discard the ones that were incorrectly typed by users. Your tool should also count the total number of PINs correctly typed by each user.

**Question 3**: Please fill in Table 3 with the total number of PINs correctly typed by each of the 30 users.   
(Total score: 10 points. Grading rubric:  
100% points for a 90~100% correctness on those numbers;  
60% points for a 60~89% correctness on those numbers;  
30% points for a 30~59% correctness on those numbers.)

Table 3 The total number of PINs correctly typed the 30 users.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User ID** | **Total Number of Correctly Typed PINs** | **User ID** | **Total Number of Correctly Typed PINs** | **User ID** | **Total Number of Correctly Typed PINs** |
| 0 |  | 10 |  | 20 |  |
| 1 |  | 11 |  | 21 |  |
| 2 |  | 12 |  | 22 |  |
| 3 |  | 13 |  | 23 |  |
| 4 |  | 14 |  | 24 |  |
| 5 |  | 15 |  | 25 |  |
| 6 |  | 16 |  | 26 |  |
| 7 |  | 17 |  | 27 |  |
| 8 |  | 18 |  | 28 |  |
| 9 |  | 19 |  | 29 |  |

### **LAB EXERCISE/STEP 4 (segment the motion sensor for individual keystrokes)**

At this point you should have a large list containing the motion sensor data of all the correctly typed PINs. You will now segment the motion sensor data for individual keystrokes (i.e., for the typing of individual digits). Improve your tool in Step 3 or write another tool to further create a list of 3-tuples. The first element of a 3-tuple contains the motion sensor data corresponding to one individual keystroke (e.g., the series of data from Line 7 to Line m in Table 1), the second element is the key or digit typed by a user, and the third element is the ID of the user who typed the key. The list will have the following format (e.g., for typing digit “5” by a user):

(([Line 7 of Table 1, Line 8 of Table 1, …, Line m of Table 1], “5”, UserID), …, ….)

Your tool should also count the total number of digits correctly typed by all the 30 users.

**Question 4**: Please fill in Table 4 with the total number of digits correctly typed by all the 30 users.  
(Total score: 10 points. Grading rubric:  
100% points for a 90~100% correctness on those numbers;  
60% points for a 60~89% correctness on those numbers;  
30% points for a 30~59% correctness on those numbers.)

Table 4 The total number of digits correctly typed by all the 30 users.

|  |  |
| --- | --- |
| **Digit** | **Total Number of Times Typed** |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |

### **LAB EXERCISE/STEP 5 (separate different types of MOTION SENSOR DATA and save the data)**

Each series of motion sensor data are composed of a sequence of 9-tuples. Each 9-tuple is recorded at the same time consisting of the acceleration forces in the x, y, and z directions, the rotation rates in the alpha, beta, and gamma directions, and the acceleration forces with gravity in the x, y, and z directions. A series of motion data (e.g., content from Line 7 to Line m in Table 1) should have the same format as shown in the Table 2. Improve your tool in Step 4 or write another tool to further separate each 9-tuple and group the same type of data for each individual keystroke. This step will help you easily extract the features from each individual keystroke in the next lab. Basically, your tool will create a long list of 3-tuples for the motion sensor data of all the 30 users, and save the data (e.g., serialize the long list of 3-tuples) to a file.

The first element of each 3-tuple is a matrix where each row is one of the 9 sequences of sensor data, the second element is the key or digit (e.g., digit “5”) typed by a user, and the third element is the ID of the user who typed the key. A 3-tuple will have the following format:

([ (accX\_1, accX\_2, …),

(accY\_1, accY\_2, …),

…,

(accgZ\_1, accgZ\_1, ...)],

“5”,

UserID)

**Question 5**: Save the processed data (e.g., serialize the long list of 3-tuples described above) to a file. If you use Python to process the data, you can include the *pickle* module to easily serialize a Python object of the processed data to a file (e.g., using "motion\_data.pkl" as the file name). Turn in this file with your complete source code for this lab.  
(Total score: 50 points. Grading rubric:  
100% points for submitting both the complete source code and the data file;  
50% points for submitting either the complete source code or the data file.)

### **Puzzler (N/A)**

This is an advanced activity for students who complete the regular activities early. N/A for this lab.

## What to submit

Please answer all the 5 questions in this lab exercise. Please feel free to directly reuse this Word document to provide and submit your answers. Please submit additional materials (if specified in an individual question) in zipped files.