smartphone Side-Channel Attacks and Defenses

**Module 1 lAB Manual**

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

**Lab Manual Contributors**:

Zhiju Yang\*, Chuan Yue@

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

# Motion sensor data collection toolkit Development

**Lab Description:** In this lab, you will build tools to collect motion sensor data from smartphone users without any restriction. You will build a website that can collect motion sensor data when a smartphone user is typing on your webpages. In more details, you will write JavaScript code, create “malicious” webpages, collect motion sensor data from the client-side, send the collected data back to a website, write server-side code to process the raw motion sensor data sent from the client-side, and evaluate the correctness of the collected data. This lab consists of six 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:

* **Design** the high-level architecture of a motion sensor data collection toolkit.
  + Assessed by the tasks and outputs specified in STEP 1.
* **Interpret** the JavaScript code for motion sensor data collection.
  + Assessed by the tasks and outputs specified in STEP 2.
* **Develop** “malicious” webpages to collect motion sensor data from the client-side and send the collected data back to a website.
  + Assessed by the tasks and outputs specified in STEPs 3 and 4.
* **Develop** server-side code to save the raw motion sensor data sent from the client-side.
  + Assessed by the tasks and outputs specified in STEP 5.
* **Evaluate** the correctness of the motion sensor data collection toolkit and the collected data.
  + Assessed by the tasks and outputs specified in STEP 6.

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

**Lab Files that are Needed:** TheLab Manual 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 (design the high-level architecture of your data collection toolkit)**

The motion sensor data collection toolkit (i.e., the website) that you will build can follow a typical client-server architecture. At the client-side, a user will visit the website using a mobile browser on a smartphone, and will type PINs displayed on a webpage. The webpage contains JavaScript code that will collect the motion sensor data corresponding to a user’s typing, and will send the data to the website at the server-side. The server-side code will save the received motion senor data to a file for further processing and analysis.

**Question 1**: Please design the high-level architecture of your data collection toolkit and draw a diagram to illustrate it.   
(Total score: 10 points. Grading rubric:  
100% points for a clear diagram and a clear description of the design;  
60% points for only a clear diagram or a clear description of the design;  
30% points for a vague diagram and/or a vague description of the design.)

### **LAB EXERCISE/STEP 2 (interpret THE JAVASCRIPT CODE FOR MOTION SENSOR DATA COLLECTION)**

This ([W3C github.io device orientation specification website](https://w3c.github.io/deviceorientation/spec-source-orientation.html)) provides the information about how to write the client-side JavaScript code to collect motion sensor data from a user’s smartphone. You can also search online to easily find many other related articles and resources. Figure 1 illustrates a piece of sample JavaScript code for motion sensor data collection at the client-side.

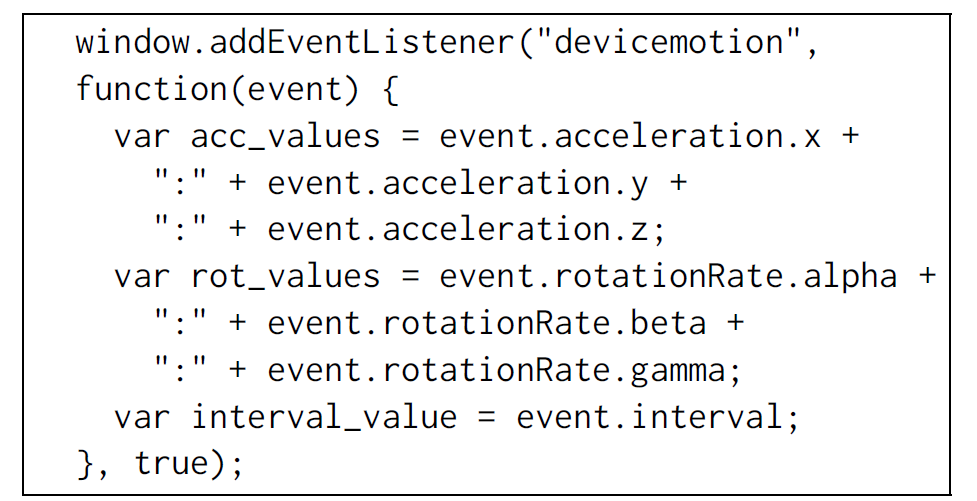


Figure 1 Sample JavaScript code for motion sensor data collection.

**Question 2**: Please briefly describe this sample code and explain what types of motion sensor data are collected.  
(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 3 (Design THE webpages FOR MOTION SENSOR DATA COLLECTION)**

You can design many webpages for your website based on the need, but one required webpage should ask a user to type something such as PINs or login information. On this webpage, your JavaScript code will collect the motion sensor data corresponding to the user’s typing.

Your webpage should meet the following requirements, so that the format of your collected data will be identical to that of the data contained in our provided dataset for you to later work on other modules:

* Each time your webpage will display a 4-digit PIN for a user to type.
* Your webpage will generate 50 unique PINs with each of them being displayed 5 times.
* Your webpage will save the motion sensor data collected from each user into a separate file.
* Your webpage will record both a displayed PIN and the PIN typed by a user, so that the typing correctness can be checked later.

Note that, you can design a virtual PIN pad on your webpage or directly let a user to type a PIN using a soft keyboard installed on the smartphone; we prefer for you to take the second approach which is simpler. Figure 2 shows an example webpage for motion sensor data collection. The webpage displays a 4-digit PIN in blue color, provides an input box below the displayed PIN for a user to type, and provides a “Submit” button for a user to submit the typing result.

**Question 3**: Design the motion sensor data collection webpage to meet the above requirements. We recommend you to use the [PHP server-side scripting language](https://www.w3schools.com/php/) to dynamically generate your webpage and send it to the client-side. Of course, you are free to use any other web development techniques to complete this step. Describe the web development techniques that you used and explain how the webpage is generated from the server-side.  
(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.)

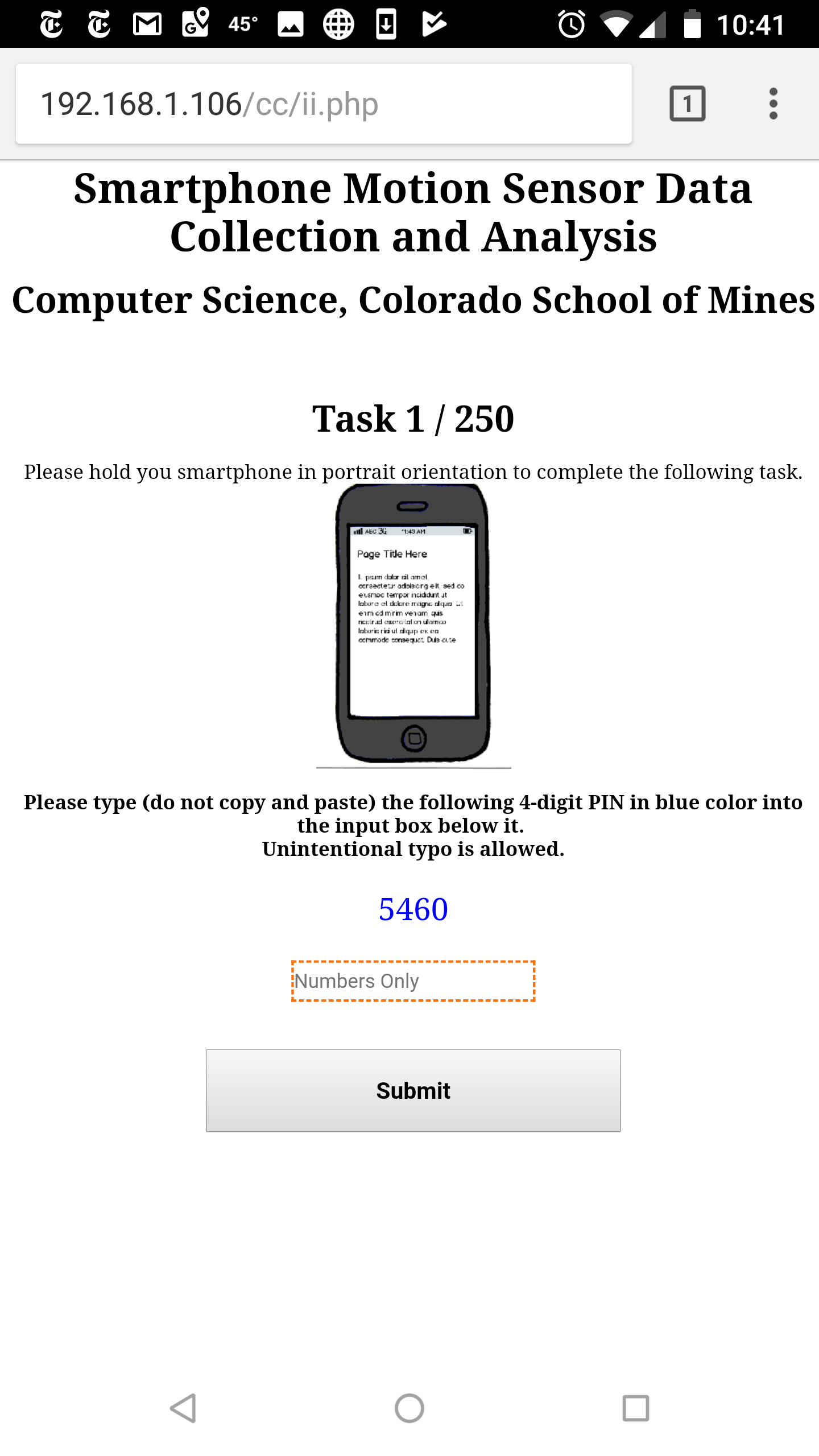


Figure 2 An example webpage for motion sensor data collection.

### **LAB EXERCISE/STEP 4 (integrate javascript code to your webpage FOR MOTION SENSOR DATA COLLECTION)**

Based on your investigation in Step 2, you need to write the JavaScript code that can collect motion sensor data and integrate your code into the motion sensor data collection webpage you developed in Step 3. To avoid collecting bulk of useless motion sensor data, you should collect the data only when a user is typing a PIN. Hint: To achieve this goal, your JavaScript code should monitor the keyboard events such as keydown, keypress, and keyup that are related to a user’s typing. For example, assume that you have the following input field:

<input id="target" type="number" value="">

You can bind a keydown event handler as shown below:

$("#target").keydown(function() {

// trigger the motion sensor data collection code

});

This [jquery.com website](https://api.jquery.com/keydown/) provides more information for you to write the code to monitor the keyboard events.

**Question 4**: Describe how you integrated your JavaScript code into your motion sensor data collection webpage, and explain how you leveraged the keyboard events in the integration.   
(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 5 (Save the MOTION SENSOR DATA at the server-side)**

In this step, you will write server-side code that can save the collected motion sensor data to some files. Your code should create a file for each user to save the motion sensor data corresponding to all the typing tasks performed by the user. Each file for a user should have 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 should be fixed (i.e., hard-coded), and they will be used for you to easily process the collected data in the follow up labs.

You are allowed to use your most familiar server-side web development techniques or languages such as PHP, Node.js, Python, or Java to create and write files, and your choice will be likely depending on what you did in the previous steps.

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 should 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 5**: Describe how your server-side code saves your motion sensor data, and explain important design and implementation details.  
(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 6 (Deploy your MOTION SENSOR DATA Collection Website and evaluate it)**

Deploy your motion sensor data collection website on your local machine or a virtual machine in the cloud such as Amazon AWS. To evaluate the correctness and attacking capability of your website, you should visit it using a mobile browser on a smartphone and check whether your website can correctly record your motion sensor data corresponding to the typing of PINs. In particular, your motion sensor data should be saved to a file following the format requirements specified in Tables 1 and 2.

**Question 6**: Describe your evaluation procedure and evaluation results. Submit the complete source code of your motion sensor data collection website with the description of its files and functionalities. Submit a file (User1Motion.txt) of motion sensor data following the format requirements specified in Tables 1 and 2; the data should be collected by your website from yourself corresponding to the typing of at least 50 PINs with each of them repeated for 5 times.   
(Total score: 30 points. Grading rubric:  
100% points for submitting all the three components including a clear description, the complete source code, and the data file User1Motion.txt;  
60% points for submitting two of the three components;   
30% points for submitting one of the three components.)

### **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 6 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.