Hardening registration

**Main Problem:**

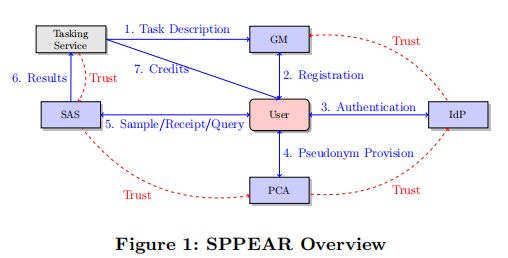
Prevent robots or malicious adversary from registering enormous amount of accounts in the system.

**Goal:**

recognize those fake “users” and stop them registering accounts.

Furthermore, prevent misbehaving devices from future attempts. Permanently evict those devices. (future direction)

SPPEAR Overview



Users register themselves to GM, which is the main focus of the project (Step 2).

**With respect to different hardened registration process, we need different approaches:**

**Rooted android devices:**

Full access permission to the file system. Therefore malicious users could manipulate information related to PS system easily. This can be tested easily when an App try to access some superuser file paths.

**Emulator:**

Detect emulator behaviors. No Bluetooth.

**Sensors:**

PS requires device sensors to work properly. Allowing only smartphones with sensors in good conditions.

**IMEI:**

Identity proof. IMEI can be accessed through Android API. Those haven’t registered in TAC database would be rejected.

**Botnet detector:**

Determine whether a device is a part of a botnet or not. Human-Machine interaction to ensure an actual human is joining the PS system.

**Location Inspector:**

Get access to the device location. Prevent those ineligible from taking part in a task which has a target location.

**Data accountability:**

An additional step to strengthen the security is introduced with the participant prompted to solve a gyroscope mini game designed in a way that the submitted

sensor data cannot be prepared beforehand.

Questions:

1: How can I get access with many accounts through rooting and spoof IMEIs?

2: Cannot understand the security and privacy part of GM properly. According to my understanding of SPPEAR system, GM have the information of users’ real identity. To prevent GM from collecting users’ willing, we limit the knowledge of GM. I learned that users could get their tokens without letting GM know. However, GM has to generate tokens which bind users’ information with tasks. Then how could GM generate these user specific tokens without knowing the details?