Updates to Design Project
6.033 2021

As mentioned in the design project spec, new requirements emerge in real-world systems all the time. Simple, modular designs can typically accommodate these changes.

To that end, we have added two additional requirements. These requirements apply to **both** design projects (exposure-tracing and MBTA). As part of the first addition, we've also clarified a bit about how the BLE network works.

1. Many of your preliminary reports used the phone's MAC address as the primary identifier throughout the system. Because a phone's MAC address *never* changes, this approach can lead to certain privacy violations. For example, if an attacker knows my phone's MAC address, they can learn a lot about me — my whereabouts, in particular — if they gain access to various pieces of data from your system.

   Solving this problem completely is difficult, so we are only asking you to address a single part of it: phones may not broadcast MAC addresses on the BLE network. They should use a different identifier — which you design — and that identifier should change over time. There is no requirement for how quickly this identifier should change. The more frequently it changes, the more robust your system is to certain attacks, but the more complex the rest of your system gets.

   You do **not** need to prove that the ID you use is cryptographically secure in any way (e.g., you don't need to prove that, given the ID, the attacker cannot easily infer the MAC address).

   Note that, while the BLE broadcasts contain only an ID, this change implies that your system is allowed to set that ID. You are not constrained to using the phone's MAC address. There was some confusion regarding that point in a few DPPRs.

   Additionally, as long as we're clarifying things about the BLE network, note that it does not provide perfect reliability. Phones may occasionally miss a broadcasted message.

2. Phones in the system may crash, and take up to ten minutes to recover and come back online. You can assume that any data stored on the phone before it crashes will still be there, uncorrupted. However, during the recovery process a phone won't be able to send or receive any messages. Again, this entire process — crash + recover — can take up to ten minutes.

   Your system should take this failure mode into account. Once a phone has recovered, are there alerts that it has missed that are still relevant? How does your system decide what is relevant, and how does it send relevant messages to the recovered phone(s)?