# *Research Statement* | ***Saumay Pushp***

This research statement describes my interest and **main works** that I have primarily **created, conceptualized, and built** during the course of my Ph.D. program. The findings of my work support corporations seeking to build accessible, universal technologies that reach diverse market segments. I publish primarily within the ACM SIGMOBILE, and SIGCHI communities.

## Current Research Overview

Every real-world applied system is built on a specific purpose to do good in our real lives. In my recent research, the main 'good' aims at **bridging the gap between device usability and privacy**, *in-situ*. Going beyond the contemporary privacy-preserving methods such as screen lock etc., in-situ privacy leaks demand a much sophisticated and subtle privacy configuring approach. While sharing the device, a device-owner has split seconds to configure a privacy setting. However, current privacy configuring methods comprises of lengthy privacy configuration steps, and therefore end-up as not being useful, in-situ. Moreover, device sharing situations occur sporadically. A person, in her daily life, comes across several people in different places. A straightforward solution, e.g., creating different privacy policies based on people or place is arguably in-feasible. My work breaks away from the traditional approaches to configure in-situ privacy settings, and explores alternative privacy provisioning systems.

• **PrivacyShield [in IMWUT/Ubicomp 2018]:** The system aims at enabling a number of unique in-situ privacy provisioning user experience. First, it provides *subtle* in-situ privacy configuration through its simple screen-less touch gestures, removing the hassle of using lengthy privacy configuration steps. Second, in-situ privacy configuration could be done in a highly *personalized* way based on the user's intention to hide (her) desired information. Third, the system provides an Application Programming Interface (API) to assist smartphone app developers in building in-situ and personalized privacy configurable applications. To realize PrivacyShield, **I developed a new screen-less touch interaction technique along with a novel stroke-based gesture-segmentation-and-recognition approach.** I leveraged the screen I/O device (screen digitizer) of smartphones. The key idea is 1) to first accurately recognize on-screen gestures that users perform even while a phone's display is *turned off*. 2) If done well, it passes the recognized gestures as gesture queries to instantly change user profiles or the system's privacy modes according to the performed gestures. As the display is off, the phone owner can pro-actively hide, e.g., pictures, notifications or kill apps, before lending or screen sharing her device. The borrower or a nearby person may not see (be aware of) what information content the owner is hiding. Users may also use gestures to subtly configure dynamic *personalized* rules, e.g., hiding message notifications from the WhatsApp app from a particular sender(s). To do so, the system facilitates individual policies associated with a gesture query.

**The core contribution of this body of work is to bring forward the need for building a technological solution to achieve privacy and device usability in the context when the smartphone devices are shared with others, intentionally or inadvertently. Importantly, I reflect upon the need for selectively hiding a subset of information, which are private to**

**the device user in a given moment, without interfering her access to the rest of the content.**

• **Chaperone [under production]:** Through my research I have dived into the most corner use-case of device usability and privacy. **I have explored the need for device privacy among the visually impaired people** (hereinafter VIPs). VIPs often seek help from people they encounter in their everyday life such as family members, friends, colleagues (hereinafter *Helpers*), by handing over their smartphones. However, due to the fact that VIPs and Helpers have different touch-screen interactions, Helpers prefer to turn-off the screen reader service on the VIPs' smartphone while providing help to easily interact with the VIP's smartphone. This makes VIPs uncomfortable as it raises several concerns: (1) privacy, (2) *Blind* to Helpers' touch actions, and (3) anxiety from losing device possession.

I conceptualized and implemented Chaperone as an easy, collaborative, and in-situ privacy-preserving help seeking service for the VIPs. The service lets a VIP to screen-share their smartphone on-screen content to a Helper's phone. Using Chaperone, the VIP first sends a request to his desired Helper in form of a message notification. Upon Helper's acceptance, VIP's smartphone screen gets shared to the Helper's smartphone. Chaperone, launches its *ATBypass-Touch-Mode* on the Helper's device to enable him interact with the VIP's shared screen without any accessibility features while not interfering with the VIP's screen-reader interface. The service provides a real-time audio feedback to the VIP on the Helper's touch actions. This allows the VIP to easily follow-up the help task and participate whenever needed. Moreover, Chaperone provides the VIP easy and in-situ privacy-preserving means, such as *On-demand UI-access Control*, *On-demand Message Notification Screening*, etc., to control Helper's access to private information in the help task. For example, using *On-demand Message Notification Screening*, the VIP can screen or block his personal notification pop-ups from a messenger app for a specific Helper. Importantly, Chaperone provides the VIP a means to collaborate during a task. For example, while seeking help on his shopping app, the VIP can use Chaperone to privately enter his credit card PIN number by pausing the session, whilst informing the Helper about his intent to do the task. **The major contribution of this work is to explore the need for trustworthy interdependence between VIPs and their Helpers using a collaborative and in-situ privacy-preserving screen sharing service.**

## Future Research Directions

In the course of addressing the problem of privacy and device usability, I developed technologies and gained important insights that can provide further benefits to smartphone users for achieving privacy on their sensitive apps. Such advantages could be derived from the novel screen-less interaction developed during the completion of PrivacyShield research. Additionally, I also obtain insights for building *surreptitious app interaction* on privacy sensitive apps which we use in our everyday lives and in front of others.

• **Designing Surreptitious Interactions for Privacy Sensitive Applications:** An interesting class of computing that could be derived from the PrivacyShield design is that of *surreptitious app interaction*. In public spaces, privately interacting with a sensitive app, e.g., a messenger app, is difficult. These public spaces are sporadically spawned by casual acquaintances who may recognize the device user along with the app he's interacting with. Some of these acquaintances, however, may not be open-minded and might objectify individuals based on their app usage. Importantly, those acquaintances are people whose opinion may affect one's work and social life.

Surreptitious app interaction would provide a smartphone user interact with his privacy sensitive app in an illusive way. I envision that an illusive app interaction could be done using a socially acceptable *decoy app* instead of engaging with the actual privacy sensitive app. For example, a smartphone user may pretend to interact with his news app whilst messaging with his friend. Realizing a surreptitious app interaction, however, has several challenges. For example, it is difficult to bring various IM-functionalities inside a decoy app while maintaining the original user satisfaction. In future, I plan to address such challenges by building IM-supported decoy apps. By building several decoy apps, we could know the feasibility and effectiveness of such apps in (a) camouflaging users' private app interaction as well as (b) their ability to support original messenger app's IM functionalities.

• **Energy Efficient and Privacy-preserving Screen-less Mobile Interaction:** Screen-less interaction capability on a smartphone is filled with several opportunities that can 1) provide privacy as well as 2) energy efficiency in plenty of the real world cases. One direction which I envision is of providing privacy protection and device power saving to VIPs. VIPs in their everyday lives are usually susceptible to visual privacy while interacting with their smartphones. The main reason is that they are unable to assess their surroundings. That is, these people find it difficult to weigh the situation where someone is curiously or intentionally looking at their screen. Also, due to their inability to scrutinize their surrounding people, they unknowingly give ample time to an onlooker to peek at their smartphone screen.

VIPs rely on voice feedback from their screen-reader-based assistive technology to interact with the GUI content on their smartphones. These assistive technologies uses a cursor which a VIP moves to read/browse different UI elements on the given GUI of the screen. However, once the device screen is turned-off, the GUI of the app and the screen-reader's cursor will get destroyed. I plan to address this difficulty by keeping an app's UI running in the background. Combining such an approach with our PrivacyShield's screen-off interaction shows promising direction in the field of providing energy efficient privacy-provisioning for a marginalized population group that happens to be prolific smartphone users.

## Overview on my Previous Research

**RAVEN [in MobiCom 2017]:** Apart from dwelling into the realms of usability-aware privacy, I have also done exciting and premiere things in regards to **optimizing power consumption on mobile devices resulting from mobile gaming applications**. Since mobile games top as the most popular class of graphics applications on smartphones and are frequently the major sources of energy drain, I built RAVEN. RAVEN is the first mobile system for optimizing rendering rate in mobile games by leveraging human visual perception. The system reduces game app's rendering rate whenever the succeeding frame is predicted to be perceptually similar enough. As a result, it optimizes the power consumption of mobile games without any modification of codes or binaries like a black magic! We set a side channel to track the rendered frame sequences which can tailor user's perception with graphics changes in a game-play. We leverage YUV color space to develop a simple, but effective way to track user's perception on-the-fly. **The main contribution of this work is to realize a never-built-before dynamic frame rate scaling solution for mobile gaming apps that is based on the human visual perception to game's graphics.**