UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

PIDE: PHYSICAL INTRUSION DETECTION FOR PERSONAL MOBILE DEVICES

Joana Rita Gaspar Velho

Dissertação orientada pelo Prof. Doutor Luís Manuel Pinto
da Rocha Afonso Carriço
e co-orientado pelo Prof. Doutor Tiago João Vieira Guerreiro

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

2015

Agradecimentos

I would like to thank to my thesis advisers Doctor Professor Luís Carriço and Tiago Guerreiro, and also Diogo Marques, for guiding me during this project. My family and friends for support and motivation. I would like to thank FCT (Fundação para a Ciência e a Tecnologia) for funding. Finally, I would like to thank all participants of the conducted studies for this thesis.

To my family.

Resumo

Os dispositivos móveis pessoais, como *smartphones* e *tablets*, permitem guardar e aceder a dados pessoais a qualquer hora e em qualquer lugar. Estes dispositivos contêm cada vez mais informação sensível sobre os seus proprietários, incluindo códigos de acesso, mensagens de texto, registo de chamadas, contactos, fotos, vídeos e informações sobre a localização geográfica. Os utilizadores parecem conscientes do risco que estes dispositivos trazem à sua privacidade. As investigações dos problemas de segurança em dispositivos móveis são, em grande parte, sobre ameaças de *software* malicioso. No entanto, uma vez que os dispositivos móveis são frequentemente utilizados na presença de outros, a ameaça colocada por pessoas próximas, fisicamente ou socialmente, tem vindo a levantar vários problemas de privacidade. Um estudo aferiu que os dispositivos móveis de 14% dos utilizadores inquiridos já foi utilizado por outra pessoa sem a sua permissão. O mesmo estudo indicou que 9% dos utilizadores confessou ter utilizado o *smartphone* de outra pessoa com a finalidade de adquirir informações pessoais.

Atualmente, o mecanismo de segurança mais comum contra intrusão física é a autenticação no ato de desbloqueio do dispositivo, seja por palavra-passe, PIN, padrão ou mesmo biométrica. Estes mecanismos de segurança são úteis quando um dispositivo é perdido ou roubado, mas ineficazes quando se trata de prevenir os amigos e a família de explorarem conteúdos num dispositivo. Os mecanismos de autenticação são vulneráveis a ataques de observação, que podem ser facilmente realizados por pessoas que pertencem ao mesmo círculo social. Por exemplo, um individuo próximo consegue facilmente descobrir um código de acesso, observando-o quando é introduzido, ou observando as marcas deixadas no ecrã tátil. Por outro lado, alguns utilizadores consideram que a autenticação é por vezes fastidiosa, já que as interações com estes dispositivos são curtas e frequentes. Por esse motivo, muitos utilizadores nunca chegam a configurar o mecanismo, ou apenas o utilizam temporariamente.

Muitas vezes, por conveniência, necessidade ou até mesmo práticas sociais, os utilizadores de dispositivos móveis são encorajados a partilhá-los com outros. Normalmente, estes dispositivos são partilhados para tarefas muitos especificas, tais como fazer chamadas telefónicas, enviar mensagens de texto, navegar na internet e até mesmo jogar. Nestas situações, os utilizadores vêm-se muitas das vezes forçados a partilhar os seus códigos de desbloqueio. Por vezes, a recusa em fazê-lo conduz a situações sociais embaraçosas,

como a demonstração de falta de confiança nos outros. Por outro lado, frequentemente os utilizadores sentem-se incomodados, e até mesmo ansiosos, com a partilha dos dispositivos, principalmente porque têm receio que os outros vejam informação privada, como fotografias ou mensagens de texto. Por esse motivo, os utilizadores mantém muitas vezes supervisão sobre as atividades realizadas por terceiros nos seus dispositivos. Mas a supervisão nem sempre é possível, o que acaba por preocupar ainda mais os utilizadores.

A autenticação no desbloqueio não dirige aos problemas de partilha dos dispositivos, uma vez que as atividades de partilha são realizadas, grande parte das vezes, após autenticação.

Posto isto, conclui-se que os mecanismos de autenticação são pouco eficazes contra oponentes com os quais existe uma relação social, seja por facilidade de acesso ao dispositivo sem o conhecimento do proprietário ou por partilha, que implica a cedência do dispositivo a outra pessoa.

Os Sistemas de Deteção de Intrusões baseiam-se no pressuposto que um sistema acabará por ser atacado. Estes sistemas são amplamente utilizados em sistemas distribuídos como uma medida de segurança que mitiga o impacto negativo das falhas de autenticação. A principal contribuição deste trabalho é a conceção e o desenvolvimento de um sistema inconspícuo de deteção de intrusões físicas para *smartphones* Android. Uma contribuição paralela é a avaliação da adequação deste sistema de deteção de intrusões, destinado a dissuadir adversários socialmente próximos de espionarem conteúdos privados do dispositivo.

Foram elaboradas entrevistas formativas *online* com utilizadores para compreender as suas preocupações e práticas comuns com os dispositivos móveis. Além disso, estas entrevistas foram essenciais para reunir um conjunto de requisitos funcionais para o desenvolvimento do sistema.

O sistema de deteção de intrusões desenvolvido executa em segundo plano e tenta determinar regularmente, através de reconhecimento facial, se o dispositivo está a ser utilizado pelo proprietário. Para isso, este sistema tira periodicamente fotografias utilizando a camara frontal do dispositivo. Caso seja verificado que o utilizador não é o proprietário, o sistema iniciará uma gravação das ações do utilizador, que podem ser revistas mais tarde pelo proprietário do dispositivo, bem como as fotografias capturadas. Sempre que é detetado uso por terceiros, para além de ser iniciada a gravação de atividades, é também lançada uma notificação informando o utilizador de que foi detetado como intruso, que as suas ações estão a ser gravadas e as suas fotografias capturadas. Esta notificação tem dois objetivos muito distintos. Por um lado, funciona como um mecanismo dissuasor que poderá impedir conduta maliciosa, como por exemplo o acesso a conteúdos que o proprietário considera como privados. Por outro lado, a notificação mitiga problemas éticos em relação aos dados privados adquiridos pela captura de fotografias e das ações do utilizador.

A principal característica deste sistema é que executa as tarefas de deteção de intrusões e gravação de interações, de forma inconspícua, o que significa que o utilizador não se apercebe da sua execução. Assim, esta aplicação torna-se num mecanismo de segurança que não requer nenhuma interação explícita.

Para concretizar o mecanismo de reconhecimento facial, utilizou-se a biblioteca OpenCV, que oferece algoritmos otimizados de deteção e reconhecimento facial, e a biblioteca JavaCV, que é uma interface em Java para OpenCV.

Para registar as ações do utilizador, foram desenvolvidos dois mecanismos de gravação distintos: *screencast* e *event-based recording*. O mecanismo *screencast* captura *screenshots*; o proprietário visualiza posteriormente as ações dos utilizadores intrusos numa sequência de imagens.

O mecanismo *event-based recording* é baseado em eventos de acessibilidade, que são mensagens lançadas pelo sistema operativo enquanto o utilizador interage com o dispositivo. Através destes eventos é possível adquirir dados suficientes para conhecer as interações que o utilizador executou no dispositivo e produzir uma lista de aplicações utilizadas e ações executadas em cada uma das aplicações.

Para validar este sistema de deteção de intrusões, foram realizados dois estudos com utilizadores. Um estudo de laboratório que tinha como objetivo, não só examinar preocupações emergentes dos utilizadores em relação à privacidade e ao uso dos seus dispositivos por terceiros, mas também identificar mecanismos de defesa e, finalmente, demonstrar a aplicação desenvolvida e compreender de que forma os participantes planeariam utilizar esta ferramenta e se a consideram útil e adequada às suas necessidades.

Posteriormente foi elaborado um estudo de campo, que permitiu aos participantes utilizarem a aplicação durante um período alargado de tempo, com o objetivo de compreender como é que os utilizadores adotaram a aplicação.

Os resultados indicam que a abordagem dos Sistemas de Deteção de Intrusões se adequa à proteção de conteúdos em situações de partilha do dispositivo e em situações em que a autenticação é insuficiente. Por um lado, funciona como um mecanismo dissuasor, por outro funciona como uma ferramenta que informa o proprietário de quem utilizou o dispositivo e com que propósito. Esta abordagem também é adequada às necessidades dos utilizadores em termos de segurança usável, nomeadamente através da oferta de uma medida de segurança que não exige que os utilizadores despendam esforço em cada interação com o dispositivo.

Palavras-chave: Dispositivos móveis, Privacidade, Segurança Usável, Inconspícuo, Adversários próximos

Abstract

Authentication mechanisms are useful when a device is lost or stolen, but ineffective when it comes to preventing friends and family from snooping through contents. Most unlock authentication methods are vulnerable to observation attacks than can easily be performed by those in a close social circle. Moreover, unlock authentication does not address the common use case of device sharing.

Intrusion Detection and Response Systems (IDRS) are based on the assumption that a system will eventually be attacked, and are widely used in network systems as an additional security measure that works around authentication flaws.

The main contribution of this work was the design and development of an inconspicuous IDRS for Android smartphones, called Auric. A parallel contribution was the evaluation of the adequacy of that approach, intended to dissuade socially-close adversaries from snooping through device contents. This system runs on the background and attempts to determine, through face recognition, if the device is being operated by the owner. If it is not, it starts recording user actions, which can later be reviewed by the owner.

We conducted a laboratory study to examine users concerns over other people looking through their data, and to present the system to participants. We also conducted a field study, where participants used the system for an extended period of time, in order to understand how they adopted it. Results indicate that the IDRS approach addresses previously unmet needs, namely by offering a security measure that does not require users to expend effort in every interaction with the device.

Keywords: Mobile devices, Privacy, Usable Security, Inconspicuous, Close Opponents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mobile devices, such as smartphones and tablets, have become ubiquitous allowing users to store and access personal data any time and any place. These devices keep data on many aspects of users' lives, such as text messages, emails, contacts, professional documents, pictures and videos. Since mobile devices are loaded with sensitive data, many users set up unlock authentication to inhibit others from accessing device contents. Modern mobile devices offer a variety of authentication mechanisms, including some based on secrets, such as PIN, password and pattern, and some based on biometrics, such as face recognition.

However, authentication raises security and usability problems. Unlock authentication is useful when a device is lost or stolen, but ineffective when it comes to preventing socially close adversaries, such as friends and family, from snooping through contents.

Unlock authentication methods based on secrets are vulnerable to observation attacks, that can easily be performed by those in a close social circle. For instance, if the opponent can observe the user when he unlocks the device, there are good odds that he/she can distinguish the key. If the opponent has continued access to the device, he/she can see oily marks left on the touch screen and discern the password pattern.

Interactions with mobile devices are usually short but frequent, which requires, deliberately and repeatedly, entering the access code each time. For that reason, some users quit or never configure unlock authentication, since they consider inconvenient to enter a code every time they want to use their devices.

For some reason, necessity or convenience, users spontaneously share their devices with others for specific tasks, such as making a phone call. Concerns about device sharing depend strongly on the level of trust between owner and borrower. Some users even share their access code with others which may compromise their privacy, but when they refuse to reveal it, social troubles may occur, such as demonstrating lack of trust in others or social embarrassment. Sometimes the owners try keeping supervision on third-parties activities on their devices, but it is not always possible, which again may compromise their privacy.

Having these issues as a starting point, we got inspiration from Intrusion Detection and Response Systems (IDRS). These systems are based on the assumption that a system will eventually be attacked, and are widely used in network systems as a security measure that works around authentication flaws. IDRSs are capable of detecting and also preventing an intrusion from successfully attacking the organization by means of an active response [60]. In a network context, an IDRS monitors a system/network searching for malicious activities. In a mobile device context against physical attacks, an IDRS should be able to identify usage by third-parties, in order to bridge unlock authentication gaps.

1.1 Goals

The main goal of this dissertation is to accomplish an effective and usable Intrusion Detection and Response System for mobile devices.

To achieve the main goal, we designed and developed an IDRS for Android smartphones that is capable of identifying usage by third-parties and responding by recording user's interactions on the device.

We conducted online formative interviews in order to define a set of functional system requirements.

We also conducted a laboratory and a field study with mobile devices users to evaluate our approach and system.

Our results indicate that the IDRS approach was found to be useful, and to cater to user's desire to have security without having to incur in constant effort and vigilance.

1.2 Contributions

The main contributions of this dissertation are:

1. Adaptation of IDRS model to the mobile devices context against physical threats.

2. Design and development of a functional Intrusion Detection and Response System for Android smartphones, which is capable of continually identifying usage by third-parties, through face recognition, and responds upon it by recording user's actions.

3. Understanding of the IDRS approach for mobile device against physical threats.

4. Publication of one research paper in a national conference:

    - Joana Velho, Diogo Marques, Tiago Guerreiro, Luís Carriço, "*Physical Intrusion Detection and Prevention for Android Smartphones*", INForum 2015 - Simpósio de Informática.

5. Submission of one research paper in an international conference:

    - Diogo Marques, Tiago Guerreiro, Luís Carriço, "*Handling physical intrusion to mobile devices with user activity logging*", Thirty-First Annual Computer Security Applications Conference (ACSAC) 2015.

1.3 Structure of the document

In the next chapter, we present the state of the art of security and privacy in mobile devices, which contains analysis of studies on users' behavior, defenses and concerns regarding security and privacy.

In Chapter 3, we present our approach of an Intrusion Detection and Response System for mobile devices and scenarios where our approach would be useful. Also, we present formative interviews conducted to collect functional requirements for our system. Also, we discuss other security and privacy proposals.

In Chapter 4, we present the system architecture, design and implementation details. In Chapter 5 and 6, we present the conducted laboratory and field studies, respectively. We conclude with Chapter 7 where we present limitations of our system, future work and other appliances.

Chapter 2

Related Work

In this chapter, we discuss concerns and practices among mobile device users reported by several studies, regarding sensitive data and usage by third parties. We also discuss protective measures that users typically take to protect their private data. Finally, we discuss similarities and differences of other security and privacy contributions.

2.1 Sensitive Data

Mobile phones are multi-functional and provide the ability to perform a wide range of actions beyond voice communication [9]. They allow users to store and have access to their personal data any time and any place. Smartphones and tablets are used for a many purposes, including to play games, access social networks, make phone calls, send text messages and shop online. With the emerging *Bring Your Own Device* trend, where employees are encouraged to use their devices to access enterprise data and systems, mobile devices also now commonly store sensitive work information [29]. A recent study [9], shows that users store data on their mobile devices that they consider sensitive, such as passwords (work related and personal), files, text messages, emails, contacts, current location information, call logs, pictures and videos. In another study [20], researchers conducted an online experiment where participants reported finding their social security numbers (20%), credit and debit card numbers (16 and 17%, respectively), bank account numbers (26%), birth dates (46%), email passwords (30%), and/or home addresses (76%) stored in their email accounts, easy accessible through their mobile devices.

Users are aware of the sensitivity of the data stored on their devices and are concerned about security threats [12, 48]. A recent large-scale study [9] reported that 70% of the participants stated that they avoid using certain functions in their device due to security concerns. The same study claims that users tend to share similar security concerns with their personal and professional data [9].

Research on security gaps in mobile devices has been leaning mostly on malware. In these attacks, the opponent does not choose the victim individually. The threat posed

by malware tends to have low impact. Indeed, the most common consequence of these attacks is unsolicited advertising [21]. However, since the personal mobile devices are often used in the presence of others, the security threat posed by people physically and socially close has recently been recognized [49].

2.2 Threat of unauthorized access

Recent studies indicate that unauthorized access by socially-close adversaries is not an uncommon occurrence and may have a particular negative impact on users. In a survey of internet users, participants reported that their mobile devices were used by someone else without permission to use its functionality (14%) and to look at some data (14%). The same study claims that 9% of participants admitted that they have used someone else's device without permission [49]. In another study [44] with young adult smartphone users, 60% of participants admitted to snoop through others smartphones contents. In another recent study, 70% of participants indicated a preference for preventing socially-close individuals from accessing some functionality on their phone [28], when confronted with a tool that provided that ability.

A common defense among users against unauthorized access is to never leave their devices unattended, for instance, keeping them close at all times, either in their pockets or purses. In a recent study [26], participants reported feel secure by physically protecting their devices despite the absence of authentication.

2.2.1 Unlock Authentication as a Defense

Users tend to use unlock authentication to protect their data in case of loss or theft. The desire to avoid family and friends from snooping through contents or past unauthorized access experiences are also often cited as a reason to lock mobile devices [20]. In a recent study, 55% of 500 participants claimed to lock their devices to prevent unauthorized access by strangers and 23% of 500 indicated that it was to avoid usage by friends and family [20].

Interactions with mobile devices are short and frequent, which requires entering the access code several times per day. For that reason, many users choose not to lock their devices or give up on unlock authentication [26]. In a recent large-scale study of smartphone users, 42% indicated that they didn't lock their devices. The most cited reason was that locking was too much of a hassle [20]. Other study, shows the same by reporting that PIN and password locks for mobile devices are not usable for most of the users, which is mainly due to their demand of an instant access to non-sensitive data or applications, such as games or Internet browsers [48].

Most unlock authentication methods are vulnerable to observation attacks, such as shoulder-surfing and smudge attacks. The most popular unlock authentication mecha-

nisms, which are based on secret codes, are susceptible to shoulder-surfing attacks, where someone could find out the access code just by looking when it is being entered [57]. Unlock authentication mechanisms, specially patterns, are also vulnerable to smudge attacks [6]. Since interactions with the touch screen leave oily residues from the fingers, an attacker can observe the marks and often infer the secret code.

In order to mitigate these observation attacks attempts to make unlock authentication unobservable have been made [9, 45], but they require too much of user's attention.

Despite authentication mechanisms are useful when a device is lost or stolen, they are ineffective when it comes to preventing social close adversaries from snooping through contents. Firstly, observation attacks can easily be performed by those in a close social circle, for instance friends and family. Secondly, device sharing allows easy access to device contents.

2.3 Threat of Device Sharing

People are prone to share mobile devices, which can not always be avoided, and privacy concerns are part of this action [38, 58]. Device sharing is spontaneous, driven by different motivations and can be initiated by the device owner or the borrower and is often limited to certain features [24]. Mobile devices users often share their devices with others for specific tasks, such as making phone calls, sending text messages and playing games [38]. As a result, sometimes the owners cannot control what others are doing on their devices, even if momentarily.

A recent study [38] found that many users are uncomfortable with guests having access to personal information, such as email, text messages, notes and files. However, privacy needs are very subjective, information that is non-personal for one person might be private for another.

In a recent study [48], participants stated that if they lend their phone to someone they know, such as friends, they would like to keep an eye on them, mainly because they feel concerned about that person looking into their data, such as messages and pictures. However, most of the time participants did not care about showing some data, such as messages and emails, to complete strangers, but did care if such data were seen by someone from their social circle. Participants from another study [58], reported having privacy concerns even when sharing the device with trusted friends, since some private data is only critical when shared with closely related people. These results suggests that the threat posed by socially close people should be taken into account in the design of privacy protection tools for mobile devices.

Sharing concerns depend on level of trust between owner and borrower. Trust may dictate if the owners keep supervision, by staying close to the borrower to observe their smartphone interaction, or not [24]. In a recent study [58], participants agreed that parents

and close friends being the most trustworthy groups. However, they disagree on data that should be shared to those groups. This suggests that sharing a mobile device is not only a matter of trust but also influenced by the data-borrower relationship .

Sharing concerns are not only related to privacy issues, but also to security issues. Smartphone owners are not only concerned about revealing private information stored on their smartphone or tablet, but they are also afraid of misbehave and unintentional misuse by the borrower, e.g. accidentally deleting data [24].

Despite these concerns, a recent study reported that is common for users to share their unlock code with others [20], which may compromise device owner privacy.

Device sharing may arise social implications. For instance, it can lead to socially awkward situations, when sensitive data is accidentally or intentionally revealed [25]. Also, reluctance in sharing the device, unlock code or some of its contents, may show mistrust to others.

Users are concerned about their data and social implications when they share their devices. They want to share only specific functionalities of their devices, while the rest should remain hidden. All-or-nothing authentication does not adequately support privacy-aware mobile device sharing [58]. It will not avoid others from snooping through contents, since it usually occurs after unlock or the borrower already knows the unlock code.

In general, users like to have full control on how the device is used and which data is accessed by the borrower [24].

2.4 New Available Solutions

There are some proposals that attempt to improve the usability of current unlock authentication methods, others address specifically the device sharing use case. Such proposals often fall somewhere between authentication and intrusion detection.

2.4.1 Fine-grained Authentication Control

Muslukhov *et al.* suggests that smartphones authentication should introduce fine-grained control, where a user can specify which application and data should be locked and which one should be accessible instantly [47]. This suggestion will not only increase security but also enhance usability, since it will reduce burden on users of constantly typing their access code for unimportant applications and data. AppLock [40] is an Android application that can lock contacts, text messages, e-mail accounts, social networks, gallery, and basically any application as a way to protect users' privacy. AppLock can hide selected pictures or videos from photo gallery, and protecting them with a PIN pad. This application suits Muslukhov's suggestion, but is still an unusable approach because a user has to authenticate every time he/she wants to access one of the locked applications. Also, it requires too much configuration effort since, for instance, a user has to select all the

contents that he/she wants to hide. Moreover, this application is also vulnerable to observation attacks. Again social implications may arise, if borrowers can see which features the owner is not sharing with them.

2.4.2 Continuous Authentication

An approach similar to ours is *continuous authentication*, in which the operator's identity is continuously monitored during the interaction with the device. Basically, these approaches were developed to minimize the inconvenient and constant explicit authentication. Some projects use user's biometric data, behavior, location or a combination of these data to calculate a confidence value. And finally, require explicit authentication to access an application or service if confidence is low.

Hayashi *et al.* [27] and Riva *et al.* [52] proposals are motivated by a trade-off between security and convenience that the mobile users are often faced. Both use geographical location, in which the user's location is used to decide whether the user should be allowed to access a certain resource by presenting an explicit authentication mechanism. They claim home, work and school as safe environments, in which the device may not require authentication. But these environments are safe considering the threat to privacy by strangers. Considering the threat to privacy posed by socially-close individuals, these approaches are not appropriate. Close social circles may have greater interest in snooping through contents than strangers.

Hayashi *et al.* [27] introduced Context-Aware Scalable Authentication (CASA). Their main idea is to choose an appropriate form of active authentication (e.g., typing a PIN) based on the combination of multiple passive factors for authentication, such as user's location, voice obtained through a microphone and correct and incorrect PIN entries. For example, if a user is at home, quick and easy explicit authentication is used.

Riva *et al.* [52] developed a progressive authentication mechanism, addressing the problem of deciding when to authenticate and for which applications. They believed that reducing the number of times a user is requested to authenticate lowers the barrier of entry for users who currently do not use any security measure. Their proposed approach combines multiple signals, such as biometric (face and voice recognition) and behavior, to determine a level of confidence in a user's authenticity, that decides whether access requires authentication.

Crawford *et al.* [16] proposed a similar approach to Riva's, in the sense that also relates confidence values to applications, but it only uses biometric data to calculate them. Specifically, Crawford *et al.* proposed a continuous and transparent authentication framework for mobile devices based on keystroke dynamics and speaker verification. This framework associates identity confidence levels to tasks. The confidence value is recalculated from biometric data acquired while devices are being used [16].

Clarke proposed NICA, which is a Non-Intrusive and Continuous Authentication sys-

tem that maintains a continuous measure of confidence in the identity of the user by gathering face, voice and keystroke dynamics data [13]. This system removes access to sensitive services and information with low confidence levels and providing automatic access with higher confidence levels. This approach only differs from Crawford *et al.* in the sense that also uses face recognition to authenticate. However, it request intrusive authentication to access a service which the user currently does not have sufficient confidence for.

Muaaz [46] proposed another continuous authentication framework based on biometrics. He proposed a multi-modal biometric system as an approach against attacks on biometrics. This framework uses three biometric techniques to identity verification of individuals in a continuous and transparent fashion: using phone sensors such as, accelerometer, gyroscope, and magnetometer; 3D-face recognition using built-in camera; and voice recognition using built-in microphones.

Kayacık *et al.* [39] and Jakobsson *et al.* [37] exploit user's behavior to authenticate by comparing current to learned behavior. Kayacık *et al.* [39] proposed a temporally and spatially aware user behavior modeling technique for sensor-based authentication. This system operates in the background and compares current behavior with a user profile. If the behavior deviates sufficiently from the established norm it triggers an explicit authentication mechanism. This system's solution automatically switches from training to deployment mode when the user's behavior is sufficiently learned.

Jakobsson *et al.* [37] developed a model of implicit authentication which has the ability to authenticate mobile users based on actions they would carry out anyway. This system authenticates users by comparing users' recent behavior to personalized models of past behaviors. For example, given that the user is in his office and has received a call from number A, then with 90% probability, he will send an email to address B within the next 10 minutes.

Almost all of these proposals authenticate the user continuously while using the device and present an explicit authentication mechanism when confidence is low. For that reason, these proposals do not address device sharing, since explicit authentication will be required if the device is being shared with another person.

2.4.3 Addressing Device Sharing Use-Cases

The following approaches address device sharing with guest views, where the borrower only can access some selected device contents. These proposals require explicit context switching.

xShare [42] allows device owners to create profiles with different rights policies. Each time the device is shared, the user has to select one of the previously defined profiles. However, this approach can lead to social implications (i.e. the borrower sees that the profile is switched) and thus, endangers the trust between device owner and borrower.

In order to prevent a child to use a smartphone improperly, for instance accidentally erase data, Windows Phone supports a tool called Kid's Corner. This tool is basically a customized guest view, that enables the access to selected applications, such as games and music, and disables the access to the rest of contents. This approach only addresses device sharing with children, but we believe that social implications may arise if the third-party is an adult.

2.4.4 Intrusion Detection and Response Systems

"*Developing systems that are absolutely secure is extremely difficult, if not generally impossible*" [18]. An intrusion detection system is a software application that monitors a networked system searching for malicious activities then reports to a management station [19]. An intrusion occurs when an attacker gains entry into or disrupts the normal operations of an information system, almost always with the intent to do harm [60]. Regarding intrusions there are four actions that a system can do: prevention, detection, reaction and correction. *Intrusion prevention* consists in taking actions to deter an intrusion. *Intrusion detection* consists of procedures that identify intrusions. *Intrusion reaction* is a set of actions taken when an intrusion is detected. These actions seek to limit the loss from an intrusion and return operations to a normal state as rapidly as possible. *Intrusion correction* is a set of actions that restore operations to a normal state and seek to identify the source and method of the intrusion, in order to ensure that the same type of attack cannot occur again [60]. When an Intrusion Detection System (IDS) detects an intrusion, activates an alarm that can be audible, visual, or silent (e.g. alert via an e-mail message). With almost all IDSs, system administrators can choose the configuration of the various alerts and the priority levels associated with each type of alert [60]. A current extension of IDS technology is the Intrusion Detection and Prevention System (IDPS), which can detect an intrusion and prevent it from successfully attacking the system by means of an active response [60].

IDS-like solutions

The following proposals were not considered by the authors as Intrusion Detection System (IDS) but they are very similar, since they are capable of detecting usage by third-parties and react upon it.

Li *et al.* [41] biometric-based system is designed to continuously re-authenticate without interrupting user-smartphone interactions. The system uses a classifier to learn the owner's finger movement patterns and checks the current user's finger movement patterns against the owner's. If patterns do not match an alert message is sent to the Operating System, that may lock the system and ask the user to input an administrator password or send a e-mail message with the current GPS information. This proposal is similar to ours

in the sense that identifies usage by third parties and never interrupts user's interactions with the device.

FaceProfiles [25] is a proposal that associates different access permissions to groups of contacts, and when a new user is detected through face recognition, permissions are recalculated and the user interface adapts accordingly by showing only applications allowed. Our approach is similar, in the sense that it also uses facial recognition to identify users and also focuses on sharing among socially-close adversaries, but differs since the reaction is not providing multi-user support, but log the users' actions. FaceProfiles requires too much configuration effort, since the user has to define policies for each group of contacts.

xShare [42] and FaceProfiles [25] address device sharing by placing the smartphone into a restricted mode. In most cases, sharing a mobile device is a spontaneous action and thus, privacy settings must adapt as quickly as possible to new sharing partners [58] to not compromising usability and to not have social implications.

Table 2.1 summarizes intrusion detection characteristics of mentioned proposals.

Solution	**Intrusion Detection**	**Intrusion Response**
Li *et al.* proposal	Finger movement patterns	Alert Message to OS
FaceProfiles	Face Recognition	Hide Applications

Table 2.1: Intrusion Detection Characteristics of IDS-like proposals.

2.5 Discussion

Concluding, authentication mechanisms are not the most appropriate way to protect mobile devices content. As previously stated, unlock authentication mechanisms may became tedious, since interactions are short and frequent, also they are susceptible to observation attacks. Unlock authentication mechanisms are intended to protect mobile devices in the event of theft or loss, but they are not appropriate to avoid friends and family from snooping through contents. There are several proposals based on the premise that smartphones must know their owner. Some proposals are based on continuous authentication using biometric data gathered inconspicuously, and when the confidence value is low they require explicit authentication. Others proposals adapt the device to a restricted mode when detect that the device is being used by someone else. Our proposal combines the continuous authentication approach, alerts in case of usage by third-party and reacts by recording user interactions, instead of entering into a restricted mode.

Chapter 3

Physical Intrusion Detection for Mobile Devices

The previous chapter reviews the current state of the art in security and privacy technologies regarding physical attacks in a mobile context. In this chapter, we describe some scenarios that may violate user's privacy. Then we present results of formative interviews with mobile device users conducted in order to explore concerns and defenses regarding usage by third-parties.

Next, we present the system requirements, based on literature and results of the interviews, and also present our proposal, called Auric, which is a Physical Intrusion Detection and Response System for Mobile Devices capable of detecting use by third-parties and reacts by recording their actions with the device. Finally, we present some scenarios where Auric is used.

3.1 Scenarios that threaten privacy

In this section we present possible scenarios that threaten user's privacy which our approach could mitigate or avoid.

3.1.1 Scenario 1: Shoulder-surfing attack scenario

Charlotte and Elizabeth are friends and expend most of their time together. Both use their smartphones to access social networks and to communicate with other friends. They both use a PIN to unlock their smartphones but they have not shared their code with each other. When they hang out, Charlotte and Elizabeth usually use their devices to post pictures of themselves on Instagram or Facebook. Charlotte has seen Elizabeth unlock the device several times that she already knows her unlock code.

One day, Elizabeth forgets her smartphone at Charlotte's home. Charlotte seizes the opportunity to copy some of pictures of herself stored in Elizabeth's device.

This scenario shows how easily unlock codes can suffer, even unintentionally, a shoulder-surfing attack.

3.1.2 Scenario 2: Smudge attack scenario

Harriot and Martin are a married couple. Both use touchscreen smartphones. Martin uses a sketch-based unlock mechanism and he never told anyone his pattern. He usually leaves his smartphone unattended at home and forgets where he left it. Harriot usually helps him finding his device and just by looking at Martin's smartphone, and she already knows Martin's unlock pattern due to the oily marks left on the touchscreen.

One day Harriot and Martin are at home. While Martin is in the shower he receives a text message. Harriot gets curious and pick up the phone, enters the unlock pattern and checks the message. Then she marks the message as unread, in order to not arouse suspicion. Harriot seizes the opportunity to check other text messages. Martin will never suspect that Harriot snooped through his text messages.

This scenario shows how easily pattern unlock codes can suffer, even unintentionally, a smudges attacks. The learned pattern unlock code may be used in later situations to violate the victim's privacy.

3.1.3 Scenario 3: Abusive Device Sharing Initiated by Borrower

Mariana is a college student and uses a smartphone for several purposes, such as access and edit her academic assignments. She uses a password to unlock her smartphone. Mariana is very careful with her device and always makes sure that no one sees her entering the unlock password. Ana is Mariana's colleague in college, whose smartphone is out of battery. Ana asks Mariana to send a text message. Mariana unlocks her device, making sure that no one sees her password, opens the text message application and lends her device to Ana. Ana is having some troubles in finish an assignment that Mariana already delivered. Desperate for help and little time to deliver, Ana seizes the opportunity not only to send the text message but also to send Mariana's assignment to herself via email. Meanwhile, Mariana is worried that Ana might read other text messages. But she doesn't supervise her colleague because it might seem inappropriate.

3.1.4 Scenario 4: Abusive Device Sharing Initiated by Device Owner

Jane and Mary are co-workers. Jane usually bring her tablet to the office and used it for several professional tasks, such as editing and consulting documents, accessing enterprise system and checking professional e-mails.

One day, Jane received a long e-mail from their boss with recommendations for a important meeting. Jane reads it and then lends her tablet to Mary so she can read it as well. Jane is very busy and does not pays attention to what Mary is doing. Mary reads the

e-mail carefully and then seizes the opportunity to check other e-mails. Mary reads an e-mail that talks about new job opportunity for Jane. Later, Mary comments what she read with other co-workers. As a result, Jane's boss discovers about that new job opportunity.

3.1.5 Scenario 5: Suspicion of Unauthorized Access

Peter is a technology lover and has many gadgets such as a smartphone, a tablet and a smartwatch. When he leaves home, Peter only takes with him his smartphone and his smartwatch, leaving the tablet at home. Peter often use authentication to unlock on his smartphone but not on his tablet, because he only uses it at home. Peter has noticed that when he returns home in the evening, the battery of his tablet decreased dramatically. Peter is suspicious that his tablet is being used by his maid cleaning during the day. While Peter is out, his maid cleaning, during her breaks, seizes the opportunity to play games on Peter's tablet.

3.2 Formative Interviews: Current Practices and Concerns

We conducted a formative interview study to understand user practices, concerns regarding possible intrusion by people in close social circles and their current defensive strategies. Also, we wanted to uncover negative experiences of physical intrusions. These interviews were also useful to collect a set of functional requirements for our system.

We opted for a loose semi-structured format, allowing the set of questions to evolve during the two-month period in which interviews were conducted. We conducted the interviews remotely, in an online instant messaging platform.

3.2.1 Participants

We conducted formative interviews with fifteen participants that use mobile devices on a daily basis, such as smartphones and tablets. Seven were female and eight were male, and ages ranges from 24 to 64 years old ($\mu = 40.9, \sigma = 12.9$). Table 3.1 presents data of recruited participants. Note that *heavy* and *light* are designations to classify device usage by participants, in which a *light* user only uses his device for phone communications and text messages, while a *heavy* user also uses it for applications, games and surf the Internet.

Participant	Gender	Age	Country	Lives with	Mobile Devices	Light or Heavy User	Lock
P1	Male	24	USA	Family	Smartphones and Tablet	Heavy	Password
P2	Female	24	USA	Significant Other	Smartphone	Heavy	Passcode
P3	Female	28	USA	Alone	Smartphone and iPod	Heavy	Passcode
P4	Male	28	India	Family	Smartphone and tablet	Heavy	Fingerprint
P5	Male	29	USA	Family	Smartphone	Heavy	Passcode/ Fingerprint
P6	Male	34	USA	Family	Smartphone and tablet	Heavy	PIN
P7	Male	39	Canada	Family	Smartphone and tablet	Heavy	No
P8	Female	43	USA	Family	Smartphone and iPod	Heavy	No
P9	Male	45	USA	Family	Smartphone, tablet	-	No
P10	Female	46	USA	Alone	Smartphone	Heavy	Password
P11	Male	46	USA	Family	Smartphone and Tablet	Heavy	Password/ Pattern
P12	Male	49	USA	Roommates	iPod	Heavy	No
P13	Female	55	USA	Family	Smartphone and tablet	Heavy	No
P14	Female	60	USA	Alone	Smartphone	Heavy	No
P15	Female	64	USA	Family	Smartphone and tablet	Heavy	No

Table 3.1: Participants in the reported exploratory interview study.

3.2.2 Analysis

The analysis of the interviews was done using thematic coding inductively [11]. The researcher that conducted the interviews compiled a set of codes by analyzing the first 8 interviews. Two other researchers then both coded a subset of 5 interviews with this book. Reliability was at an acceptable level for this subset, average Cohen's $\kappa = 0.85$. Hence, one of the researchers coded the remaining 10 interviews.

3.2.3 Results

Concerns

We identified general concerns users have about their personal mobile devices related to usage by third-parties in their social circles. Some participants reported not keeping data they would consider private in their devices. Yet, sensitive data is not the only concern, for those people, there are still worries about misuse:

> "I would be worried that they [family members] would do something to mess it up, but there is really nothing that I would have a problem with anyone seeing. I do not handle any finances, do any work or anything really important on it." (P9)

> "I have about five games that I am in the middle of. I have them in a separate folder and told them [other people that sometimes use the device] not to play those games." (P12)

We observed a variety of concerns over unauthorized access to sensitive information. While some participants report not being concerned over their own devices being snooped on, they projected the threat onto their close ones, as, for instance, P9:

> "For my kids, I would not want people having access to their websites, photos and other data they have on their phones."

We confirmed what had already been reported in another study [24], that concerns about sensitive content are, even among people in close social circles, highly person-dependent. For instance:

> "If a family member or a kid had taken my phone, I would feel more concerned. If it's a friend, then not so much." (P4)

Most of participants classify their devices as highly personal. Therefore, they are concerned over privacy invasion, even when the device is only momentarily handled by others. Even if any particular threat can be articulated, participants reported feeling anxiety about device separation:

> "I always get uneasy when someone has my phone even for a little bit and almost sort of start to grab back for it." (P2)

> "[when someone else is using my device I feel] protective, maybe a little tiny bit nervous. Even if you trust the person not to, you can't control whether or not they go poking around in some other app or something." (P3)

Notwithstanding the anxiety of sharing their mobile devices, users still let others handle their devices due to social standards. For instance:

> "I've given my phone to people to use. [...] I can't say no if someone needs to make a call. It feels wrong." (P10)

Also, we observed that the expression of excessive control or denying access to a third-party may be seen as inappropriate. Indeed, users face a conflict between full device control and socially-acceptable behavior.

Practices

We also tried to understand what mobile device users usually do to protect themselves against unauthorized access and device sharing. Some participants were skeptical about unauthorized access for three main reasons: lack of interest from potential attackers, lack of skills and trust. Some participants reported that they do not do anything in particular to protect themselves, because no one would be interested in what they keep on their devices. For instance:

> "My sister asked me to delete some texts recently from a conversation we had about my son. It was kind of insensitive [about a personal issue]. I don't think he would snoop anyway. Yes, I don't think that they [the children] think they would nd anything interesting." (P15)

Others reported that people, who could have physical access to their device, such as children and older family members, lack the ability to do so. Specifically, some potential attackers are not a threat simply because they would not know how to use the device. For instance:

> "They [the children] like their own games and shows so would not "wander" into anything else. Plus, they would have a hard time understanding other things." (P8)

Some participants reported that their relationships are trustworthy, and for that reason, they do not feel necessity in protect their mobile devices.

> "I trust all of those who are close to me. My wife and children would not make purchases. They have their own phones as well. I trust they have no interest in accessing my banking, or online accounts. I trust that my friends and coworkers have no interest or desire to access my device. I realize that is naive, but I can't imagine a coworker picking up my device." (P7)

> "I don't snoop, so they don't either. My brother would ask before he looked at anything else on the phone. My mother wouldn't know how to snoop on the phone – [but] she would maybe try." (P10)

Not only the type of relationship, but also the kind of device is strongly related with socially-imposed access rules. Generally, the smartphone is seen as highly personal, in opposition to other mobile devices:

> "Well, my phone is usually on me, but my tablets are just laying around in the house. My wife does not have to ask [for permission to use them], but kids will." (P9)

> "I just don't let my cell phone out of my sight /hands / purse when I'm outside my house. My iPod never leaves the house but I also do not save passwords on any personal apps so I have to log in each time on those." (P8)

Keeping mobile devices physically close is usually used as a way to prevent unauthorized access. But this behavior may vary depending on the environment and trust on relationships. For instance:

> "When I am outside of my home my devices are always in my pockets or in my view. When I am at home I don't get too far from them but they have lied around. I am not too concerned with the people in my home causing me trouble." (P6)

> "I trust the people at work and at home and the phone is locked, and if I'm out somewhere the phone is most likely in my pocket or in my purse."(P2)

However, trust relationships seem insufficient for some participants that reported effort to keep the devices under control, regardless the environment and trust. For instance:

> "I never leave them [other people] alone with the phone." (P5)

> "When I take my girls to their gymnastics class, I will leave my purse and ask a friend to keep an eye on it. [I] will leave my wallet but will not leave my phone." (P8)

As mentioned previously, a common practice today, is device sharing. Some participants reported that usually show other people content on the device, like pictures. In these situations, participants usually not let others hold the device as a defense against the threat of unauthorized access to some contents. For instance:

> "If I show a picture it's not handed over, and no-one but me or my wife uses my phone/tablet. Looking doesn't require it to be held." (P11)

> "If I want to show someone something (a photo, for example), I hold the phone where they can see it." (P14)

Finally, another defense practice, reported by many participants, is to avoid keeping sensitive data on their devices as a preventive practice:

> "I also think we have to accept some responsibility for what is on our phones and devices. I try to be careful about what I keep on all my devices, even my Kindle." (P10)

> "I also don't put anything in a text or online that I'm unwilling to have become public." (P14)

One participant reported to actively cleaning data considered sensitive from his device.

> "I usually delete my texts after I read them, and for any pictures that I do not want to share with anyone, I delete them after saving it in my Dropbox." (P4)

These results suggest that current security measures are not sufficient, since users feel need to protect themselves in other ways, such as keeping devices close or maintain supervision when sharing their device.

Negative experiences

Participants reported negative experiences which suggests that common defensive practices are not always effective mostly due to social standards. For instance, P10 reported that, despite keeping the device close, people have used it unexpectedly, where negative reaction could have social implications:

> "At work during lunch I often have my phone on the table. People have walked by and said 'oh is that the 5C or 5 or is that the 6?' [then] grab and swipe. I just would never think to pick up something so personal. People have emails and photos and texts. It is really uncomfortable." (P10)

Despite social pressure, participants reported trying to reclaim the device when others were using it. For instance:

> "There have been times when people were 'just looking' at it and it made me anxious. So I have jokingly wrestled it back from them! I'm sure I came across as pathetic, but it isn't just a phone anymore. If someone has your phone they now have your email, your photo album, your banking info, your apps, your recent purchases, books you've downloaded, videos you've watched. Not just a phone. I'm not paranoid, honest!!" (P10)

> "After a staff meeting at work, I had an image to share with a colleague. He had a laugh, and others wanted to see so it got passed around the table. About 3/4 of the way around, I announced: okay, gimme my phone back. There was no inappropriate or embarrassing content on my phone. In retrospect, must be due to the possibility that my wife might send a racy or inappropriate text message." (P7)

Two participants reported having to reclaim the device in situations that were expanding towards privacy invasion, and even so had to do it jokingly:

> "I was talking [messaging] to my ex-girlfriend and she [my cousin] took my phone and I felt uncomfortable as I didn't want to share the conversation with anyone else. I was messaging with her. She [the cousin] snatched it from me as joke [but I got it back soon after]" (P4)

> "One of my brothers actually opened up my texts. He quickly shut the app tough when I asked what he was doing. I [...] made him choke on his drink by suggesting that if he kept going he would be seeing some naughty photos. None there, just a threat, [but] no brother wants [to see] that!" (P10)

These reports suggests that, despite the anxiety of having a third-party using their device, users tend to reclaim their devices without evidencing lack on trust or damaging existing social relationships.

Receptiveness to IDRS

Finally, we tried to perceive if users would find it useful if their device recorded suspicious interactions, that could be reviewed later. Most participants saw usefulness in maintaining some sort of activity log. Some reported to past experiences to explain it:

> "I have left my phone behind, forgotten it places before and had to go back. Knowing if someone had accessed it would be helpful, especially since there is such sensitive data on there." (P10)

> "It would be interesting to see what people would do if I left it unattended." (P2)

We also asked about types of things that should be recorded. Participants' suggestions including recording of the intruders face; timestamped logs of applications used, what was looked at during usage periods, and account sign-in attempts; location traces; full video replay of the suspected activity, showing the whole screen; audio recordings of phone calls

3.3 Auric

Based on the assumption that a mobile device will eventually be used by a third-party with or without owner's permission, we designed and developed Auric.

Auric is a physical Intrusion Detection and Response System for Android smartphones that empowers users' security and privacy. Our system is capable of identifying if the device is being used by a third party and responds by recording the actions performed on the device.

3.3.1 System Requirements

Based on literature review and results from the formative interviews, such as users' concerns, practices and suggestions, we defined a list of desirable requirements.

Functional requirements

- **Detect use by third parties:** The system should continuously authenticate the user in order to identifying if the device is being used by others and react upon it.
- **Recording intruders face:** The system should capture intruders face.
- **Recording users interactions:** The system should register accessed applications or record a video of what is showing on the screen.

Non-functional requirements

- **Usability:** The system should be user-friendly and easy to use, as well as easy to set up, because too much effort to set up may lead to withdrawal of use. Also, it should allow spontaneous device sharing.
- **Availability:** The system should always be in a functioning condition.
- **Transparency:** The system should be transparent and do not disturb the regular usage of the device, i.e. operating in background.
- **Modifiability and Extensibility:** The system should be modular in the sense that can easily incorporate new features and easy to modify.

3.3.2 System Description

We designed and developed an inconspicuous Intrusion Detection and Response System for Android smartphones. It can prevent, detect and react to intrusions. It is capable of identifying if the device is being operated by a third-party and reacts by recording user actions. The recordings are made available for later review. It offers the device owner the opportunity to know *who* used his device and for *what* purpose.

Intrusion detection

We design Auric to continually authenticate using biometric characteristics, since it does not require the user to perform any specific or explicit action. We decided to continuously authenticate the user through face recognition, gathering data using the front-facing camera. For that reason, Auric only runs on devices with built-in frontal camera.

Specifically, while the user is interacting with the device, Auric is, periodically and inconspicuously, taking pictures using the front-facing camera and processing a face recognition analysis. This way the user can operate the device normally while the system runs on the background.

The user is only successfully authenticated if the face recognition matches the owner's face with a certain confidence value, which means that if no face is detected, it is regarded as an intrusion. For instance, if an attacker covers the front-facing camera with his thumb, an intrusion is considered to be underway.

Other biometric measures could be used in this system. Yet, gathering data using camera, such as pictures or even video, has an additional advantage which is showing to the device owner *who* used his device. Using other biometric measures, such as keystroke dynamics or touch analysis, it would be impossible to identify the attacker, if Auric was not trained for that.

Moreover, another biometric measure could be used to authenticate alongside taking pictures. Furthermore, multi-modal intrusion detection mechanism could be used in our system. For instance, a consensus algorithm could be developed to decide, using data from touch analysis, keystroke dynamics and face recognition, if an intrusion is underway.

Intrusion reaction

This system's reaction to an intrusion is to record user's actions, in such a way that those recordings can be later audited by the owner. Alternatively, Auric can record all interaction regardless of the intrusion detection outcome. In this case, the intrusion detection results will only be used to filter device owner's recorded activities.

Auric supports two different ways to record intruders actions by recording a video of user interactions, or by capturing events of user interactions and presenting a time-line of used applications.

A *false positive*, in other words, absence of an actual attack, will only introduce an extra recording. If Auric is recording all interactions, regardless of the outcome of the intrusion detection, a *false negative*, in other words a failure of detecting an actual attack, will be recorded as well and has no or limited negative impact.

Intrusion prevention

Surveillance cameras protect people, places and objects by constantly monitoring physical spaces. It is well know that just the awareness of their existence inhibits misbehavior [14]. Mirroring this concept, we propose making the device display a permanent warning (as a notification), informing the operator that pictures will be taken and actions on the device will be recorded. In a close social context, even if the attacker hides his/her identity to the camera, it is likely that the owner is able to perform the identification given other context (e.g., time and even the details of the attack).

Modus Operandi

System's *modus operandi* is based on the *session* concept, which is a period of time that the device is interactive, specifically between an unlock and re-lock. Auric starts its work when the device becomes interactive and stops when the device becomes non-interactive. Specifically, when the device wakes up the intrusion detector starts and records depending on its outcome. When the device becomes non-interactive, i.e. when it is locked, the intrusion detection and recording will stop.

3.4 Auric's Impact in Scenarios that threaten privacy

In this section, we present the impact of Auric in the scenarios presented previously.

3.4.1 Scenario 1: Shoulder-surfing attack scenario

Elizabeth and Charlotte scenario may seam harmless because Charlotte only used her smartphone to gather some pictures of herself, but it is nonetheless an abusive attitude, since she used her device without permission. If Elizabeth had Auric installed on her device she would be able to see what Charlotte did in her absence. Or if Charlotte was aware that her pictures and actions were being recorded, she might have asked her friend before sending those pictures.

3.4.2 Scenario 2: Smudge attack scenario

Remembering Harriot and Martin's scenario, if Auric was installed on Martin's smartphone he could easily discover that Harriot was snooping through his text messages by checking Auric's recordings. Or Harriot would not snooped if she was aware that Martin will discover.

In this scenario, Auric might have solved the problem by inhibiting Harriot's misbehavior.

3.4.3 Scenario 3: Abusive Device Sharing Initiated by Borrower

Remembering Mariana and Ana's scenario, if Mariana had Auric installed on her smartphone, she would be able to verify that her colleague Ana had sent the assignment to herself. Or if Ana was aware that her pictures and actions were being recorded she might not have sent that file to herself. Otherwise social a implication would arise. Even if Ana, for example, covered the camera with her thumb, Auric would record her actions as well. If Mariana checked Auric's recordings immediately, she would know, without any doubt, that Ana stole her assignment. Or even if she checked later, by the time, she could easily realize that those actions were performed by Ana.

In this scenario, Auric may solve this problem by inhibiting Ana's misbehavior or mitigate it by informing Mariana.

3.4.4 Scenario 4: Abusive Device Sharing Initiated by Device Owner

Remembering Jane and Mary's scenario, if Jane had Auric installed on her device, and if Mary was aware of it, she might not snooped through Jane's e-mails. Otherwise Jane would know.

3.4.5 Scenario 5: Suspicion of Unauthorized Access

Remembering Peter scenario, if Peter had Auric installed on his tablet he could discover that his maid is using his tablet to play games without permission and threatening his privacy as well.

3.5 Discussion

Auric was developed in parallel with the interviews and incorporates almost all user's suggestions. Auric allows device owners to acquire knowledge of who used his device and for what purpose, i.e. the owner will be informed of which data were accessed and changed. Yet, our approach can work as a prevention of misbehavior, just as surveillance cameras do. Third-parties may be aware that their actions are being recorded and their pictures are being taken. Consequently, Auric also works as a deterrence method to inhibit close social circles to snoop through device contents. If a user is aware that his/her actions are begin recorded, he/she might not misbehave.

Currently, our approach does not actually inhibit a third-party to access or change device contents, which means that does not protect confidentiality or integrity of data stored. Yet, Auric is capable of detecting third-parties, and for that reason, a feature could be added that inhibits access to sensitive data. However, our approach seams to be powerful in the sense that it inhibits misbehavior without complex authentication mechanisms that require a lot effort to the user. Instead, it just appeals to moral values and alerts the user to the social consequences that unauthorized access to the device or piece of information or functionality, may have.

Chapter 4

Architecture, Design and Implementation

In this chapter, we present Auric's architecture based on its runtime behavior. We introduce system's architecture components and how they communicate with each other. Finally, we discuss data storage decisions.

4.1 System Architecture

Our system is composed of three main components: a *Visualization UI*, a *Data Repository*, and the *Auric Service*, which communicates with the Android Framework. The *Visualization UI* is used for presenting data to users. The *Data Repository* is used for a data-oriented connection, where the *Auric Service* and *Visualization UI* components compute concurrently and communicate via a shared repository using specific connectors, such as SQLite queries, key-value pairs of simple data types in a shared preferences file, and files in Android's file system.

4.1.1 Auric Service Components

The *Auric Service* component has three main components: the *Intrusion Detection*, the *Recording* and the *Service*. The *Intrusion Detection* and the *Recording* components are independent and coordinated by *Service* component. The *Intrusion Detection* component is responsible for detecting usage by third-parties, for that reason it is responsible for taking pictures, processing face recognition, deciding if an intrusion is occurring and storing intrusion information. This component uses a built-in frontal camera to capture users pictures. The *Recording* component is responsible for recording user's interactions. It uses an accessibility service to acquire data about users interaction and then process and store information about user's actions on the device. And finally, the *Service* component is responsible to manage the other components, i.e. starts and stops recordings depending on intrusion detection result. Also it starts and stops components when the device becomes

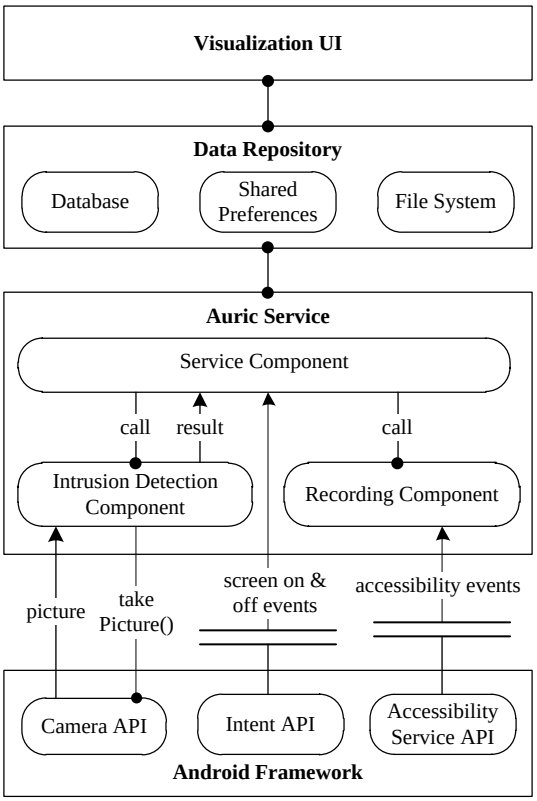


Figure 4.1: Overview of the System Runtime Architecture.

interactive and non-interactive, respectively. Figure 4.1 presents an overview of Auric's architecture.

The *Service Component* is notified when two types of Android system events occur. It listens to "screen on" events, that are sent when the device wakes up and becomes interactive; and "screen off" events, that are sent when the device becomes non-interactive. When a user starts interacting with the device, the *Service Component* starts the *Intrusion Detection Component* which in turn periodically sends messages indicating if an intrusion was detected or not. To decide if an intrusion is occurring, the *Intrusion Detection Component* communicates with the *Android Camera API* to take pictures inconspicuously that go through a facial recognition analysis. Depending on that intrusion detection results the *Service Component* prompts the *Recording Component* to start or stop. If an intrusion is detected, the *Recording Component* will start recording user's interactions. If the owner is detected the *Recording Component* will stop if it was running.

Service Component

The *Service Component* component contains a *Service Thread*, a *Message Queue* and a *Receiver*. The *Service Thread* communicates with the *Intrusion Detection Component* and

Recording Component through method calling to start and stop their jobs. The Receiver listens "screen on" and "screen off" events, and add a message to the *Message Queue* corresponding the event received. The *Message Queue* also receives the results sent by the *Intrusion Detection* component. All messages are processed by the *Service Thread*. Figure 4.2 depicts the *Service Component*.

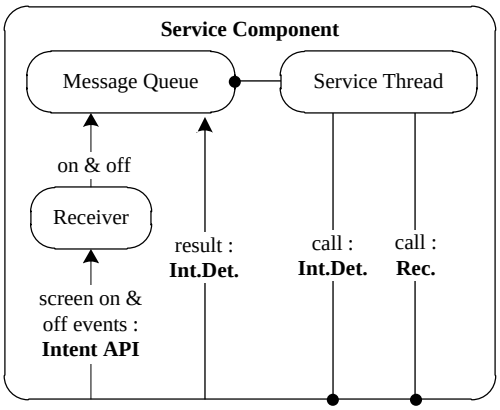


Figure 4.2: Service Component details.

When a user starts to interact with the device, the *Receiver* receives an event and sends a message to *Message Queue* informing that an interaction with the device has began. The *Service Thread* reads that message, prompts the *Intrusion Detection* component, and waits for a result. If the device owner is not recognized, the *Service Thread* will prompt the *Recording* component to be activated. If the user is recognized as device owner, the *Service Thread* will prompt the *Recording* component to stop.

The *Service* component acts as a coordinator of the *Intrusion Detection* and *Recording* components. Therefore, these components remain independent from each other. This component is also responsible for launching a notification informing the user that a intrusion was detected.

Intrusion Detection Component

This component is responsible for capturing pictures, processing face recognition analysis and deciding if the device is under a possible intrusion or not. Specifically, this component includes *Intrusion Detector Thread* that awakes periodically and, through method calling, demands the *Camera Manager* to take a picture inconspicuously using the front-facing camera. After receiving the picture taken, the *Intrusion Detector Thread* communicates with the *Face Recognition* component through method calling, asking for a facial recognition analysis. This module performs face preprocessing, detection and then recognition. If this analysis indicates that the picture taken does not match the owner's face, or matches

with a low level of confidence, a possible intrusion is considered to be underway. Finally this component reports the result to the *Service Component*. Figure 4.3 depicts the *Intrusion Detection Component*.

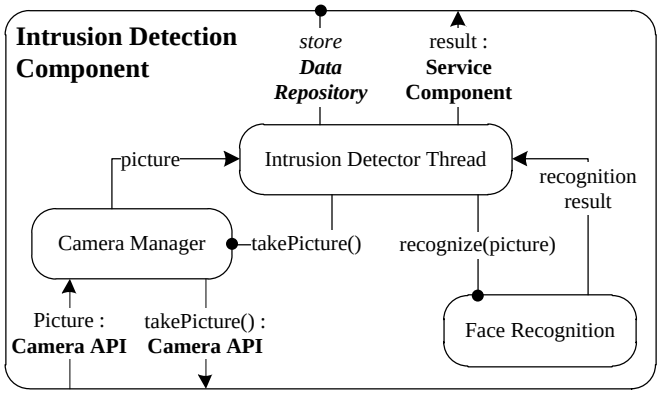


Figure 4.3: Intrusion Detection Component details.

Recording Component

This component is responsible for recording user interactions on the device. It gathers and processes data to a representation that is suitable for auditing. It supports two different methods of recording user interactions: the *screencast* and the *event-based recording*. Figure 4.4 depicts the *Recording Component*.

The *Recording Component* has a *Screen Recorder* that performs the *screencast* method, which records a video of what is showing on the screen.

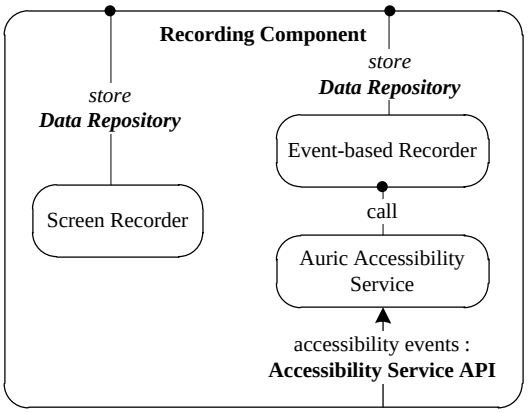


Figure 4.4: Recording Component details.

The *Recording Component* also has an *Event-based Recorder* which records user interactions using the data from accessibility events. Accessibility events are messages about user interactions with visual interface components in an application. Those events help to produce a detailed log of user interactions with the device. The *Auric Accessibility Service* listens specific accessibility events, such as text changing, view clicked, view selected, view scrolled and others. When one of those events occur, the *Auric Accessibility Service* is notified and sends that data to the *Event-based Recorder* to be processed and stored.

The *screencast* method produces a video of user's activities on the device. Each frame of the video is a screenshot taken while the user is operating the device.

The event-based recording method produces a list of applications accessed and details about user's interactions in each application. This method relies on events provided by Android's accessibility API.

4.1.2 Data Repository

In this section, we discuss data storage decisions.

The principal data storage options in Android are saving key-value pairs of simple data types in a shared preferences file; saving files in Android's file system; and saving structured data in databases managed by SQLite [30].

Data to Store

- **Event-based Recordings:** List of acceded applications and interactions perfomed within each application.
- **Screencast:** A set of screenshots (PNG images).
- **Intruders Photographs** A set of pictures captured using the front-facing camera.
- **Session Data:** A set of attributes, such as date, time, type of recorder, intrusions associated and others.
- **Face Recognition Data:** A face database and an auxiliary file.
- **Owner Pictures:** One picture of each individual trained to be recognized as device owner. These pictures will only be used by the user interface, to show to the user all subjects trained as owner.
- **Preferences:** A set of preferences, such as, intrusion detector chosen (only Face Recognition available), operation strategy chosen (record everything or record only intruder's interactions), recording method chosen (screencast or event-based) and passcode if it is defined.

Shared Preferences

We choose to use `SharedPreferences` APIs to save preferences, such as, recording and intrusion detection methods. This data storage option consists on saving key-value pairs of simple data types in a shared preferences file. This is the best data storage option in this case because is a relatively small collection of key-values to save. A `SharedPreferences` object points to a file containing key-value pairs and provides simple methods to read and write them. Each `SharedPreferences` file is managed by the framework and can be private or shared, in this case we kept it private [32].

Database managed by SQLite

Saving data to a database managed by SQLite is ideal for structured data, such as session data and event-based recordings. Databases are only visible to the application who create them and contents can be accessed and updated using queries, reducing the complexity of the application code [31, 51].

For each session, there are several pictures associated corresponding to the person who operated the device. Those pictures are saved with half the original size and for that reason, we decided to store them in a database managed by SQLite. The Figure 4.5 presents Auric's SQLite tables.

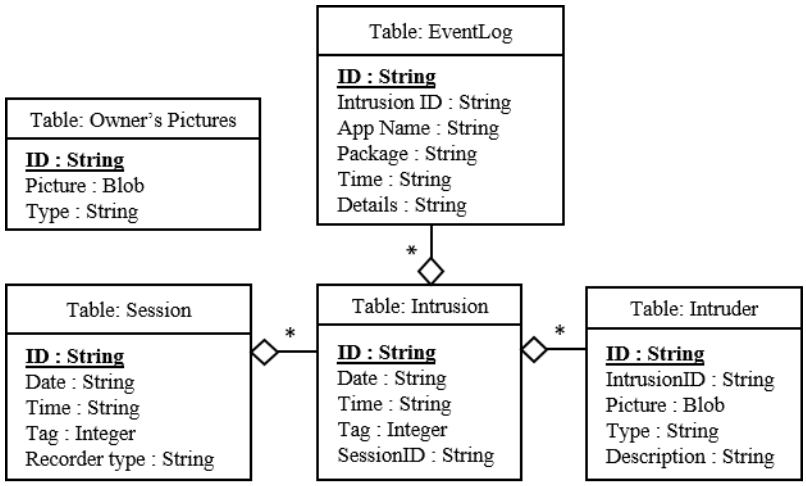


Figure 4.5: Auric's SQLite Tables.

Files in Android's File System

Recording data generated by the screencast recording method, which is a set of screenshots, are stored it in files in application's external private directory that can only be accessed by the application. Screenshots taken are too large to be stored in SQLite database which has 1MB limit [55].

OpenCV and JavaCV methods require the face database and the auxiliary file to be stored in Android's file system. For that reason, we decided to store that data in Auric's external private directory. The Figure 4.6 presents an overview of Auric's external private directory.

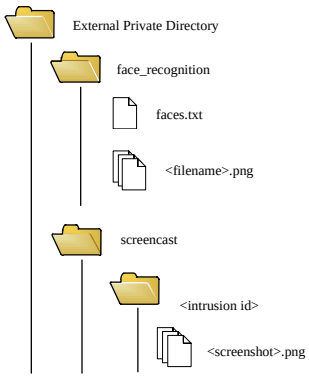


Figure 4.6: Auric's external private directory overview.

4.2 Design and Implementation

We design our system by separating concerns and constructing independent modules that address different aspects, allowing evolution by replacing modules and also boosting understanding and reuse.

The creation of independent modules improves system maintainability and extensibility, in such a way that it becomes easy to add new features, such as new intrusion detectors, recording methods and operating strategies.

Auric is composed for ten main packages: `service`, `strategy`, `record`, `accessibility`, `detector`, `camera`, `recognition`, `data`, `activities` and `utils`. Figure 4.7 presents an overview of Auric's packages.

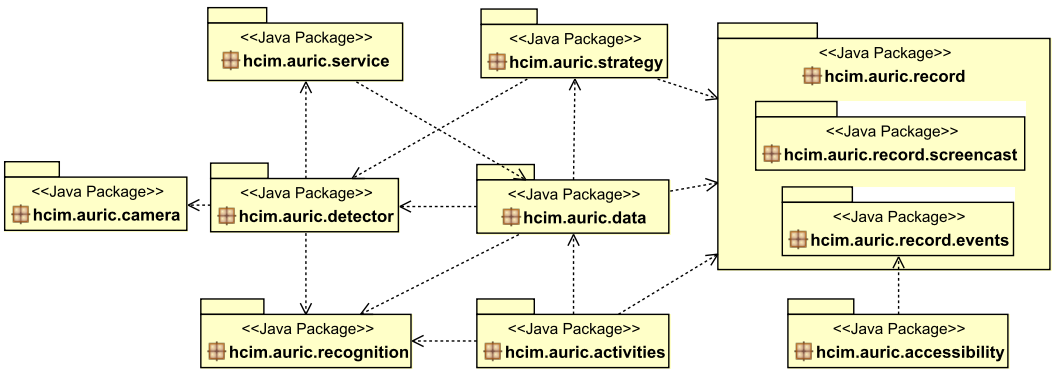


Figure 4.7: Auric's Package Overview, without `utils` package.

Each package is a module with specific responsibilities. The `activities` package contains Java classes responsible for interacting with the user. The `data` package is responsible for storing data, either in databases or in shared preferences file. The `detector` package is responsible for detecting intrusions and was designed to facilitate the addition and modification of intrusion detectors. The `record` package is responsible for recording user interactions and was designed to facilitate the addition and modification of recording methods. The `recognition` package is responsible for processing face detection, training, recognition and preprocessing. The `service` package is responsible for managing the background service. The `strategy` package is responsible for coordinating Intrusion Detection and Recordings. The `camera` package is responsible for encapsulating specific methods related to Android `Camera` APIs. The `accessibility` package is responsible for encapsulating specific methods related to Android `AccessibilityService` APIs.

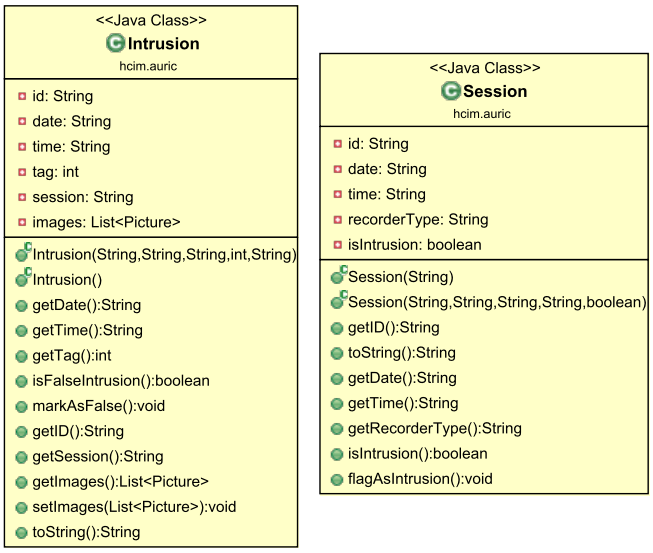


Figure 4.8: `Intrusion`, `Session` Java Classes.

`Session` Java class represents the device usage between it became interactive until it became non-interactive again. `Intrusion` Java class represents the portions of a session that were considered that the device was being used by a third-party (Figure 4.8).

4.2.1 Service Module

Basically, the `service` package is responsible for managing the background service that coordinates intrusion detection and recordings (Figure 4.9).

Android development environment allows the creation of background services by providing an abstract class called `Service`. The `Service` class represents an application's desire to perform a longer-running operation without interacting with the user, or to sup-

ply functionality for other applications to use. The application asks the system to schedule work for the service, to be run until the service or someone else explicitly stop it. [35].

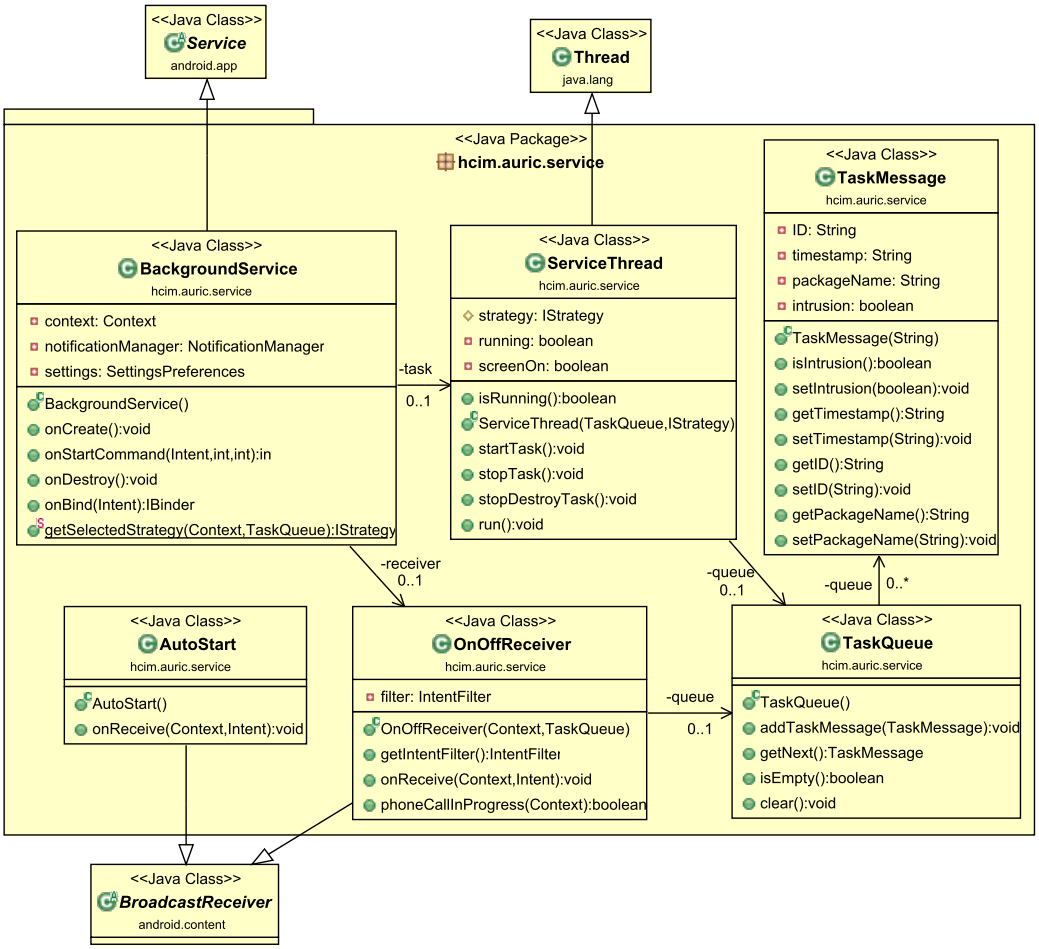


Figure 4.9: `service` package overview.

The `Services` are similar to other application objects, in the sense that they run in the main thread of their hosting process. For that reason, a service that is going to do intensive operations should create its own thread to execute those operations.

The `BackgroundService` is an instance of `Service` abstract class, and the `ServiceThread` is the thread that executes intensive operations. In this case, the `ServiceThread` coordinates intrusion detection and recordings depending on a strategy. Operating strategies are implemented in the `strategy` package. We decided to implement strategies independently so it is not necessary to modify service related code to create a new coordination strategy.

The `ServiceThread` has a `MessageQueue`, which is a queue of `TaskMessages`, that contains messages send from other components, such as the intrusion detection results from the `IntrusionDetector` and events from the `OnOffReceiver`.

The `BackgroundService` has a `OnOffReceiver`, which is `BroadcastReceiver`, that listens *screen on* and *screen off* events, which represents device becoming

interactive and non-interactive, respectively. When the `OnOffReceiver` receives those events send a `TaskMessage` to the `MessageQueue` to be processed by the `Service-Thread`.

The `AutoStart` is also a `BroadcastReceiver` that listens the *boot complete* event, which is broadcasted after the system has finished booting. Upon receiving this event, the `AutoStart` starts the `BackgroundService` if it was running before shutdown.

4.2.2 Strategy Module

The `strategy` package contains the Java classes that implement different strategies. A strategy is a plan of action depending on the outcome of the intrusion detector (Figure 4.10). Strategy is an abstraction and was created to easily accommodate new operating strategies without having to modify the classes from `service` package.

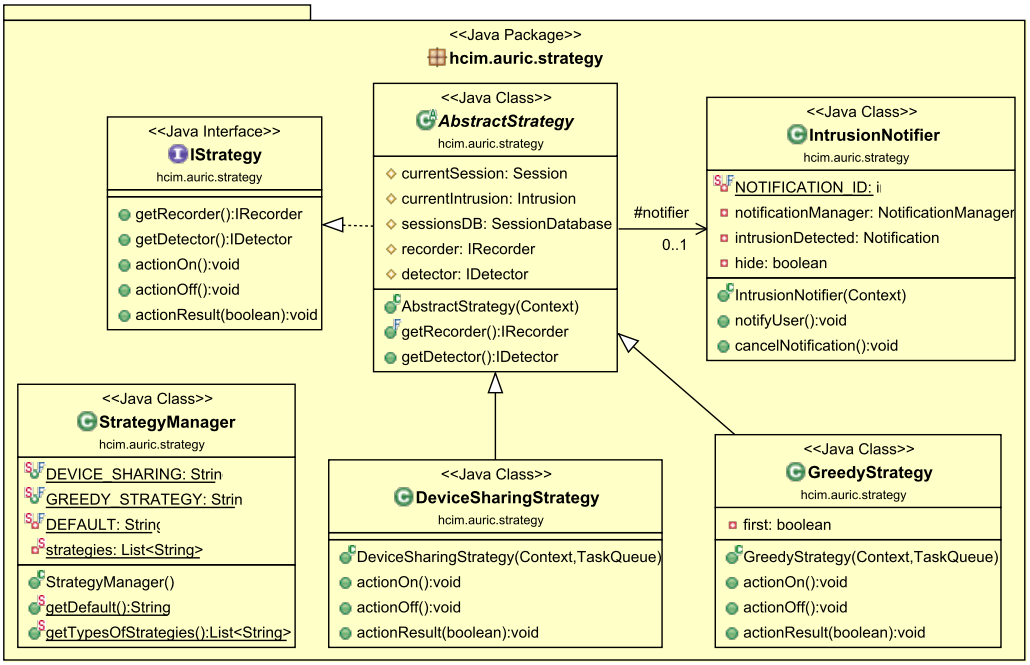


Figure 4.10: `strategy` package overview.

The `IStrategy` is an interface that allows the creation of multiple strategies without interfering with other parts of the system. The `AbstractStrategy` is an instance of `IStrategy` and has a skeleton implementation common to all strategies implemented.

Auric supports the `DeviceSharingStrategy` and the `GreedyStrategy`, both are instances of `AbstractStrategy`. The `DeviceSharingStrategy` only records interactions of an intruder, i.e. if a third-party is operating the device, this strategy starts recording interactions. If in the meantime the owner is detected, the recording stops. While the `GreedyStrategy` records all interactions regardless the intrusion detection

outcome. Both of these strategies, `DeviceSharingStrategy` and `GreedyStrategy` are independent from recording and intrusion detection method.

The `StrategyManager` is responsible for managing different strategies. It has a list of the supported strategies and informs which strategy was selected by the user.

4.2.3 Recording Module

The `record` package is responsible for recording user interactions (Figure 4.11). The `IRecorder` is an interface that allows the creation of multiple types of recorders without interfering with other parts of the system. The `RecorderManager` is responsible for managing different recorders. It has a list of the supported recorders and informs which recorder was selected by the user. The `record` package has two additional packages that separates the two supported recording methods: the `record.screencast` and the `record.events` packages.

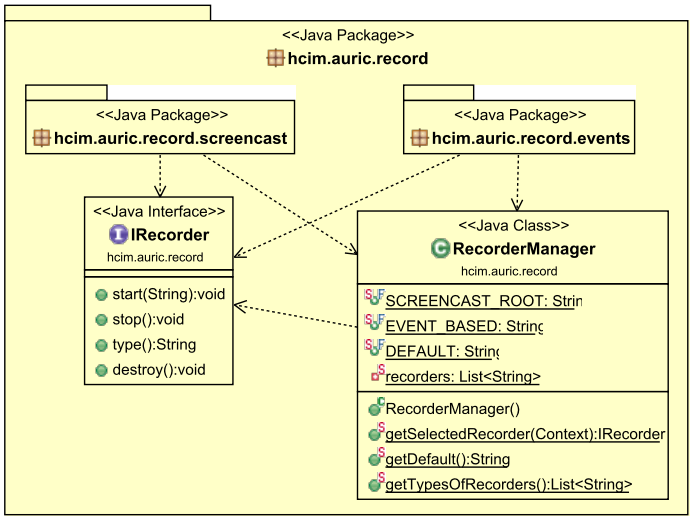


Figure 4.11: `record` package overview.

Screencast Recording Module

We start developing the *screencast* method which captures screenshots while the user is interacting with device. Then presents that data to the device owner as a video. The `record.screencast` package is responsible for recording user interactions using this method (Figure 4.12).

The `ScreencastRecorder` is and instance of `IRecorder`. It is responsible for starting the `RecordScreen`, which is a thread that takes screenshots, while the user is interacting with device, and stores those images in Auric's external private directory, where any other application can access.

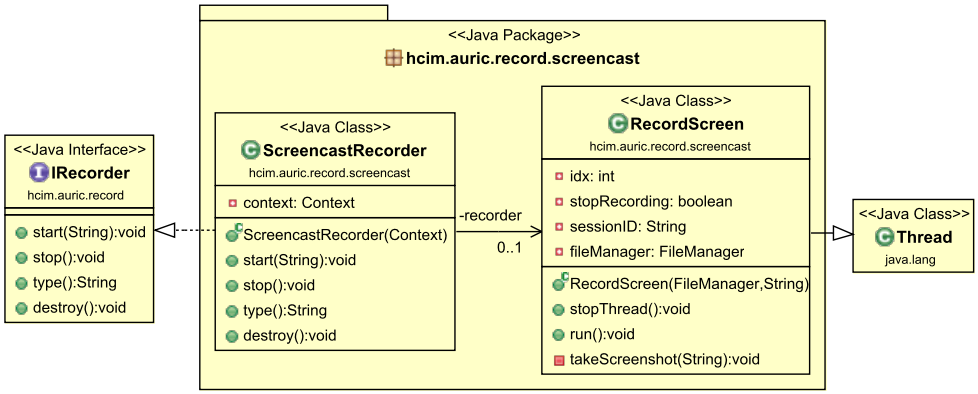


Figure 4.12: `record.screencast` package overview.

Taking a screenshot programmatically in Android requires executing a command with administrator privileges (`su /system/bin/screencap -p <filename>`). Therefore *Screencast Recording* method requires that the target device has root or administrator permissions.

The Android system is based on a Linux kernel, which means that it works with the same levels of permissions for users, files and features. When a device leaves the factory, it comes out with some limitations that are imposed by manufacturers for the sake of security and stability of the operating system. Is not common to have root permissions in a mobile device, and to enable a large and differentiated population of participants, a new recording method that could run on mobile devices without special permissions was needed.

There are many applications available that record what happens on the screen, such as RERAN[22] which is a timing-sensitive and touch-sensitive record and replay for Android smartphones. EverTutor[59] is another application to record what is showing on the screen. It was developed for creating interactive tutorials on Android smartphones. Both applications require root permissions.

We attempt develop a recording method that captures what is showing on the screen without requiring root permissions on the target device. The intent was to develop a method with a similar approach to Recordable Android application[43].

Recordable is a screen recorder for Android with advanced features such as audio recording and gesture rendering. This application uses Android Debug Bridge (ADB) to record a device screen without root permissions[43].

Google code released a library, the Android Screenshot Library (ASL) [15] that provides means for taking screenshots of phone's screen without the need for having privileged permissions. ASL utilizes background native service which performs screen capturing on demand from an application that uses the library. This service has to be started using ADB, which means that the background native service provides screenshot-taking functionality for as long as the phone is not rebooted [15, 33].

We tried to develop a *screencast* method using this ASL, but this attempt failed. ASL supported by Google Code did not work on the devices for testing. Also, this library has not recent updates, last update was on January 2011, and may be is deprecated. Even if we had succeeded, this approach would not be usable, because the users must connect their devices to a computer every time their device boots, in order to enable the recording feature via ADB.

Event-based Recording Module

`record.events` package is responsible for recording user interactions using the *event-based recording* method (Figure 4.13).

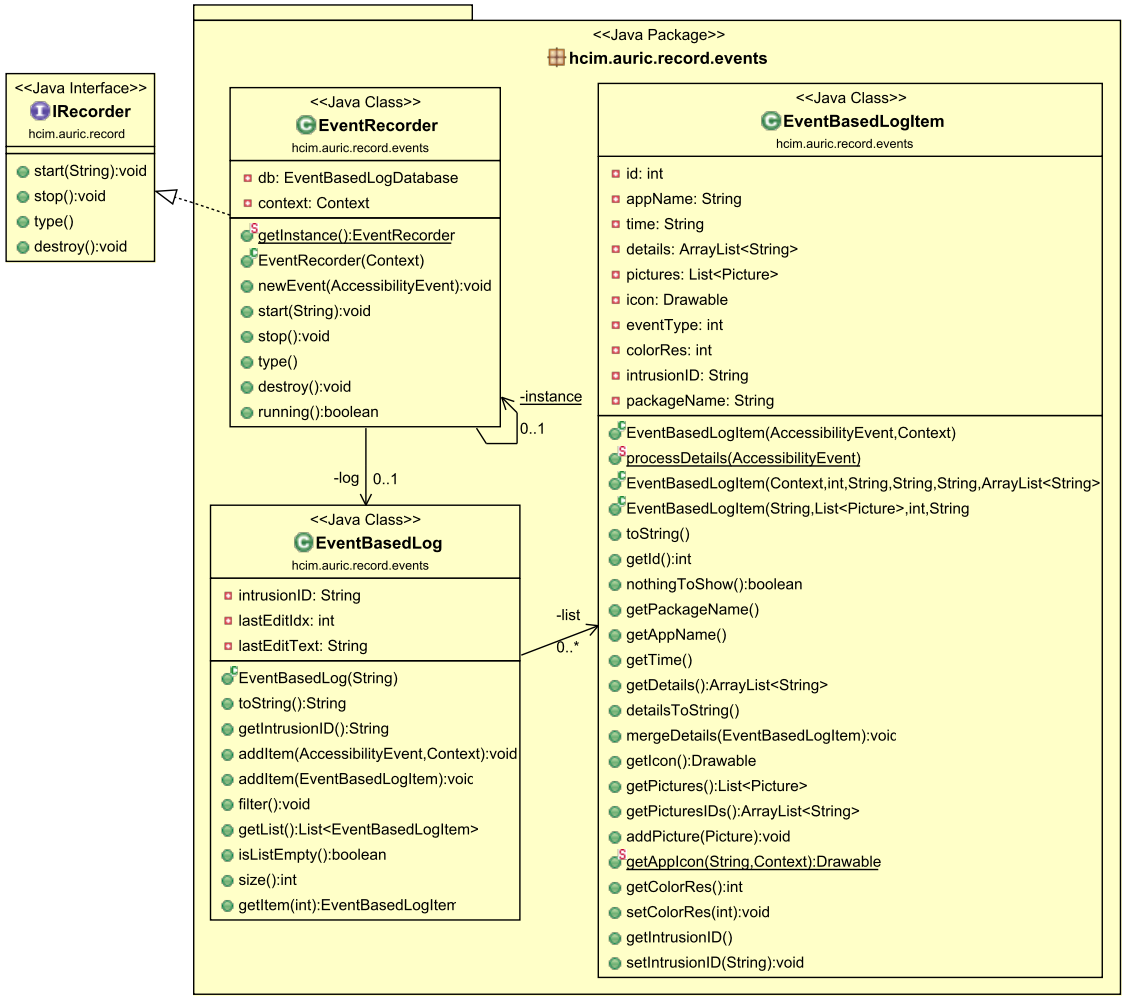


Figure 4.13: `record.events` package overview.

The failure of developing a video recording method that does not require administrator permission led to the design of the *event-based recording* method.

The *event-based recording* method produces a list of applications accessed and details about user's interactions in each application. This method relies on events provided

by Android's accessibility API. Accessibility events are messages about user interactions with visual interface components in an application. Those events help to produce a detailed log of users interactions with the device. The `record.events` package communicates with an accessibility service that listens to specific accessibility events, such as text changing, view clicked, view selected, view scrolled and others. When one of those events occur, the accessibility service is notified and sends that information to this module to be processed and stored.

The `EventRecorder` it is an instance of `IRecorder` and is responsible for recording user's interactions using accessibility events received through the `AuricAccessibilityService`. Specifically, it is responsible to process and store data from accessibility events. Each application accessed is represented by the `EventBasedLogItem` which contains user interactions performed within that application. `EventBasedLog` is a list of `EventBasedLogItem`, and represents all applications accessed within an intrusion.

Gotcha! [4] is an Android application that has a similar to the *event-based recording* method. This application allows users to know if their device was lifted and what applications were launched. Yet, it does not use accessibility events and does not offer details about user interactions performed in each application launched (Figure 4.14).

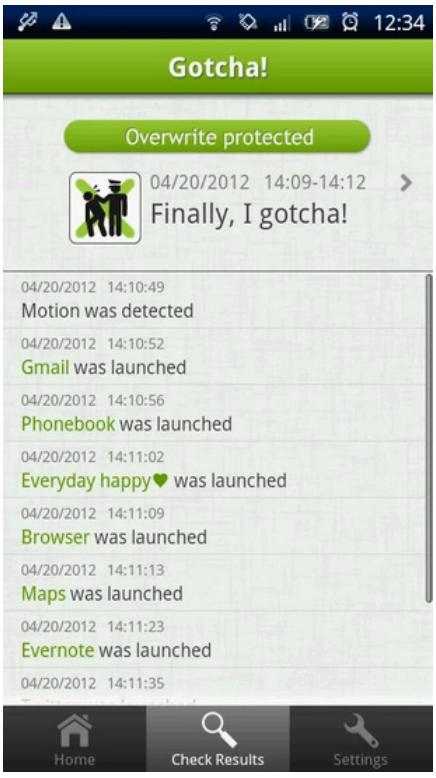


Figure 4.14: Gotcha! Android Application log review [4].

4.2.4 Accessibility Module

In `EventManager` Java class are defined the events that will be captured and processed. Auric captures *view clicked* and *view long clicked* events, that indicate which interface components were clicked; *text changed* events that indicate which text entries were changed and what was written; *view selected* events that indicate which interface components were selected; *view scrolled* events that indicate that a scroll gesture was performed; and *notification state changed* events that indicate that notification was launched.

`accessibility` package is responsible for communicating with `AccessibilityService` API (Figure 4.15).

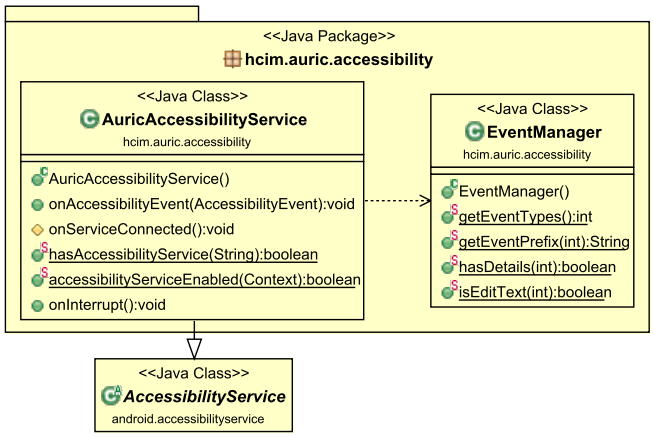


Figure 4.15: `accessibility` package overview.

The `AuricAccessibilityService` is an instance of `AccessibilityService` abstract class, and it listens the events defined on EventManager and forwards them to the `EventRecorder`. Figure 4.16 presents the process of recording user interactions using the *event-based recording* method.

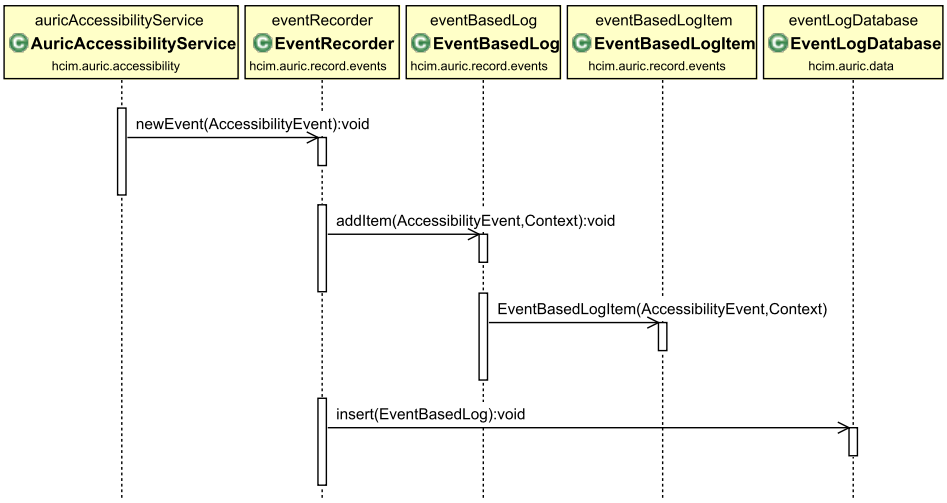


Figure 4.16: Event-based recording process.

4.2.5 Intrusion Detection Module

The `IDetector` is an interface that allows the creation of new types of intrusion detectors without affecting other parts of the system.

The `IntrusionDetector` is an instance of `IDetector` that detects intrusions by checking if a third-party is operating the device using face recognition. This class has an `IntruderCapture` which is a class responsible for taking pictures periodically by communicating with camera package. The `IntrusionDetector` is also an observer of the `FrontPictureCallback` from `camera` package that awaits for the taken picture. When a picture is taken, the `IntrusionDetector` is notified. Which in turn demands a facial recognition analysis to the `FaceRecognition` class in the `recognition` package. After that the `IntrusionDetector` stores the picture taken and the respective face recognition result. Finally, the `IntrusionDetector` sends a `TaskMessage` with the intrusion detection result to the `MessageQueue` from `service` package.

The `DetectorManager` is responsible for managing different intrusion detectors. It has a list of the supported detectors and informs which detector was selected by the user.

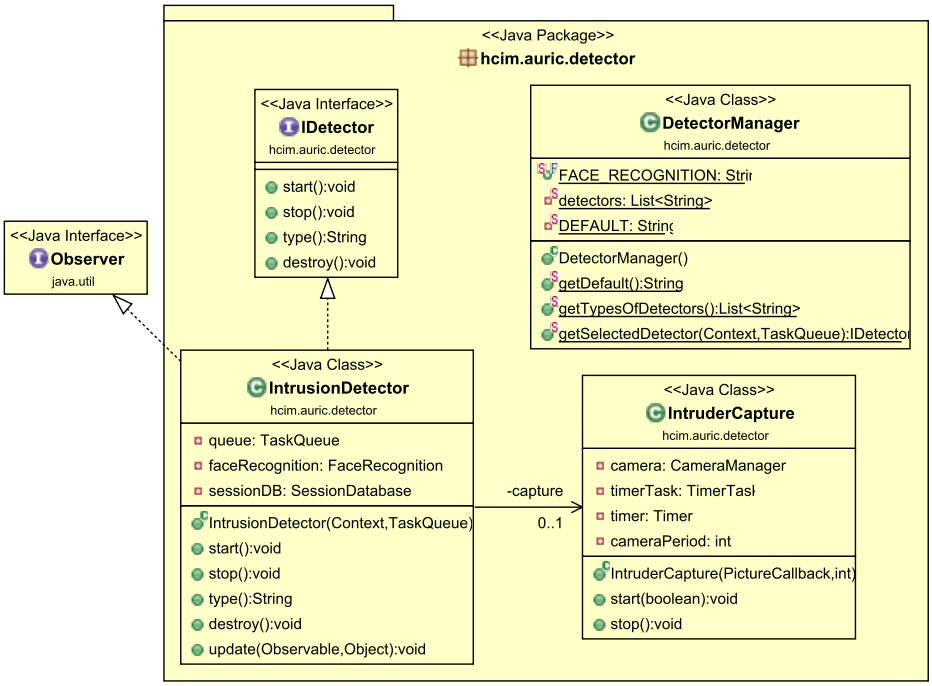


Figure 4.17: `detector` package overview.

4.2.6 Camera Module

A decisive design decision in this project was to choose what kind of data would be captured by the camera, i.e. a choice between a video against a set of pictures. On one

hand, a video recorded from front-facing camera during all interactions would help to contextualize the device owner. On the other hand, that feature will lead to inability to use the front-facing camera hardware by other applications, which can only be use by only one process at a time. Also, if the video had audio source has well, it would be stopped by an upcoming phone call. Moreover, it would have greater impact on battery consumption. For that reason, we decided to take pictures periodically, since it offers information about who used the device and uses less times the camera and have less impact on battery consumption.

The `camera` package is responsible for communicating with the `Camera` API in order to capture pictures inconspicuously (Figure 4.18). This package contains a `CameraManager` which is responsible for preparing the camera, such as setting camera parameters, and triggering an asynchronous image capture.

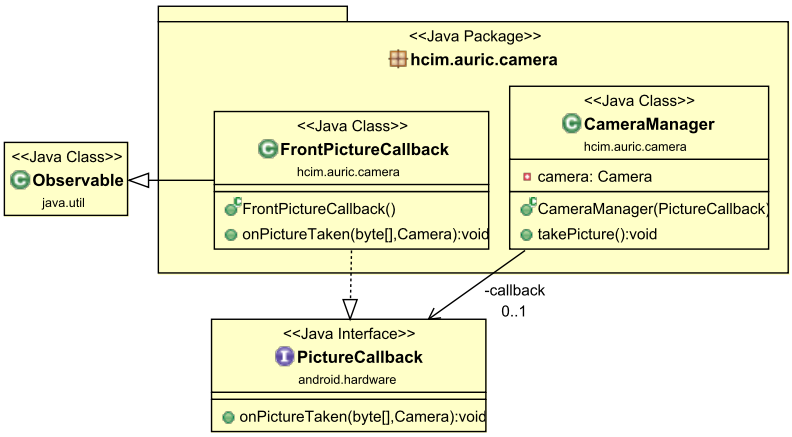


Figure 4.18: `camera` package overview.

When a picture is taken, the `FrontPictureCallback` receives it and notify its observer, the `IntrusionDetector` from the `detector` package that a picture was taken. The `IntrusionDetector` and the `FrontPictureCallback` have an observer/observable relationship which means that the `FrontPictureCallback` does not have to know which objects will receive the taken picture. This way `FrontPictureCallback` could be reused for other purposes. Also, the `FrontPictureCallback` is responsible for disconnecting the camera, otherwise the camera will remain locked and be unavailable to other applications.

Taking pictures inconspicuously requires no visual feedback that a picture is being taken. Android documentation states that taking a picture usually requires that users see a camera preview before the capture[34]. The `SurfaceTexture` is Java class that captures frames from an image stream as an OpenGL ES texture. Using the `SurfaceTexture` as a preview texture allows capturing pictures without visible preview. For that reason, the `CameraManager` uses an empty `SurfaceTexture` as a preview texture for camera. Empirical evidences show that this technique does not works properly on all

devices. Table 4.1 contains a list of tested devices.

Device	Android	Takes pictures inconspicuously
Samsung Galaxy S4	4.3	Yes
Samsung Galaxy S3	4.3	Yes
Samsung Galaxy S2	4.4	Yes
Samsug Galaxy Grand Duos	4.3	Yes
Samsung Galaxy Nexus	4.3	No
Samsung Google Nexus S	4.1	Yes
Wiko Getaway	4.4	Yes

Table 4.1: Smartphones used in inconspicuous picture capturing test.

4.2.7 Face Recognition Module

Humans perform face recognition routinely and effortlessly in their daily. Since very young, humans learn to recognize their family and friends faces [36].

Automatic face recognition is all about extracting meaningful facial features from an image, save them into a useful representation and performing some kind of classification on them [50]. Basically, face recognition is a process of matching a given face to a known face. For a computer to learn to recognize a known face, usually takes four main steps: face detection, preprocessing, training and recognition.

OpenCV (Open Source Computer Vision) is a computer vision library focused on real-time image processing and includes patent-free implementations of the latest computer vision algorithms [50]. We choose to use the OpenCV Library because it supports a SDK for Android and tools for developing face detection and recognition applications. Also, we used JavaCV, which is a Java interface to OpenCV which allow us to implement some features in Java instead of C++.

Face Detection Algorithm

To detect faces OpenCV[10] library offers a Cascade Classification. The OpenCV 2.4 version comes with various pretrained XML detectors for different purposes, such as face, eye, mouth and nose detection. For this project, we wanted to detect frontal faces, since the application captures users' pictures using the front-facing camera, so we decided to use the Local Binary Pattern (LBP) face detector, because it is the fastest face detection algorithm supported by OpenCV [7].

Face Recognition Algorithm

The difference between face detection and recognition is that face detection is determining if there is some face in the image, but in face recognition is determining whose face it

is. OpenCV v2.4 provides three face recognition algorithms [50, 7]: Eigenfaces is also referred to as Principal Component Analysis (PCA), it was first used by Turk and Pentland in 1991 [56]; Fisherfaces is also referred to as Linear Discriminant Analysis (LDA), it was created by Belhumeur, Hespanha and Kriegman in 1997 [8]; and Local Binary Pattern Histograms (LBPH) was created by Ahonen, Hadid and Pietikäinen in 2004 [1].

All three aforementioned algorithms perform face recognition by comparing a given face with some training set of known faces, searching for a match. Yet, these algorithms use the training set a bit differently. Eigenfaces and Fisherfaces find a mathematical description of the most dominant features of the training set as a whole. LBPH analyses each face in the training set independently [3].

A study compared PCA, LDA and LBP algorithms and results clearly showed the superiority of the LBP-based recognition (83.9% versus 76.3% and 69.5% for PCA and LDA, respectively) [23]. For that reason, we decided to used this algorithm to perform face recognition.

Specifically, Local Binary Patterns Histograms (LBPH) algorithm analyzes each image in the dataset independently. It analyzes the images is by characterizing the local patterns in each location in the image. When a new unknown image is provided, it performs the same analysis on it and compare the result to each of the images in the dataset.

A face recognition result has two attributes, which is the name of the matching face and the difference between a given picture and its match. Therefore, a user is successfully recognized as device owner if and only if his/her pictures matches device owner's trained faces with a small difference value.

Faces Database

To perform the face recognition algorithm it is necessary to create a database of faces. A database of faces contains a set of different images of each distinct subjects.

OpenCV Library suggests to use some databases, which we chose AT&T face database [54]. The AT&T face database contains 10 different images of each of 40 distinct subjects. For some subjects, the images were taken at different facial expressions and details, for instance with and without glasses, and different light conditions. All the images were taken against a uniform dark background with the subjects in frontal position [54]. We only used 16 subjects from this database, because when Auric trained the AT&T face database, we found that some of the images were not successfully trained due to failure of detecting a face. Therefore, we discarded all subjects whose at least one image had not been successfully trained. Which means that only 16 subjects had their ten images trained successfully.

Auric's enrollment takes 10 pictures of the device owner and trains them. Therefore, our system has at least 170 face images in its face database, 10 from the device owner and 10 from each of 16 subjects from AT&T face database. These facial images are saved in

Android file system.

Training and Recognize Procedure

Face Training is the process of saving all preprocessed faces for each person that should be recognized, and then learning how to recognize them. Face Recognition is the process of checking which of collected faces are most similar to a given face. Before training or recognize, a facial image go through three processes:

1. **Image preprocessing** It is the process of converting a color image to grayscale and then apply histogram equalization which is a process that standardizes the brightness and contrast of a facial image.

2. **Face detection** It is the process of locating a face region in an image. This step does not care who the person is, just that it is a human face.

3. **Face preprocessing** It is the process of cropping the image to just show the face region.

After these steps the facial image is prepared to be trained, in order to be recognized later, or suffer a face recognition analysis, in order to find a match to a trained face.

We performed an empirical evaluation to measure how long it takes to process the facial recognition analysis, which includes preprocessing steps, face detection and recognition. We ran Auric's face recognition analysis in 150 pictures of a subject, on a Samsung Galaxy S4 smartphone. Results show that our system takes on average $\mu = 1301.9$ milliseconds to process face recognition with a standard deviation $\sigma = 131,0$.

Measuring Face Recognition Accuracy

Many research papers show face recognition accuracy rates above 95%, but when testing those same algorithms, accuracy often is lower than 50%. This comes from the fact that current face recognition techniques are very sensitive to exact conditions in the images, such as the type and direction of lighting and shadows, exact orientation of the face, expression of the face, and even the current mood of the person. For example, if a person was standing to the left-hand side of the lights in a room when training, and then stood to the right-hand side while testing face recognition, it may give quite bad results[7].

For that reason, we conducted an empirical evaluation of face recognition accuracy where we used 150 pictures of three different subjects with different face expressions and light conditions, in order to test resilience to false positives and negatives.

We asked a female subject to set up Auric with her pictures which collects 10 pictures. Therefore, we have 10 pictures of the device owner plus 10 pictures of each 16 subjects from AT&T face database.

In order to measure false positives, i.e. frequency that the system does not recognize the owner and assumes that an intrusion is underway, we processed Auric's face recognition analysis in 150 pictures of the same subject. 95,3% of 150 pictures matched owner's face, 76,7% matched owner's face with a small difference value and 18,7% matched owner's face with a large difference. This means that the owner was successfully recognized as owner in 76,7% of 150 pictures.

4,7% of 150 pictures did not match owner's face, which means that either matched other subject's face from the AT&T face database or did not match any trained subject.

In short, results show that our system produced 23,3% of false positives, i.e. failed recognizing the device owner, since 4,7% did not match owner's face and 18,7% matched owner's face with a large difference.

In order to test Auric's resilience to intruders, we assume a worst-case scenario where the intruder is very similar to the device owner. Therefore, we used test camera activity to gather face recognition results of 150 pictures of first subject's twin sister.

Results show that our intrusion detector, in the worst-case scenario, produced 52% false positives, i.e. considered an intruder as device owner. Only 48% of 150 pictures were considered pictures of an intruder. Which means that a similar person can be easily be recognized as device owner. However, the number pictures of device owner's twin that did not match owner's face are superior (28%) comparing with the device owner result (4,7%). Yet, the percentage of pictures that matched owner's face with a large difference is similar, specifically 20% for twin sister and 18,7% for device owner.

Then we test Auric's resilience to intruders assuming the average-case scenario where the intruder is different from device owner. We process a face recognition analysis in 150 pictures of a third subject. In this case, Auric produced 22% false negatives, i.e. considered the intruder as device owner. Auric succeeded 78% considering that an intruder was operating the device, whereupon 70% of 150 pictures did not match owner's face and 8% matched owner's face with a large difference.

These results prove that face recognition sometimes fails. Auric produced 23,3% of false positives, 22% and 52% of false negatives in the average and worst-case scenario, respectively. Auric is not sufficiently accurate when two persons are too similar, since it fails most of times. Also it proves that different light conditions and face expressions affect face recognition accuracy for false positives about 28% of the time.

Measuring Face Detection Resilience

Note that the previous analysis only covers face recognition accuracy, face detection can also be affect by light conditions and face orientation. For that reason, we performed a face detection in 200 pictures of a subject with different light conditions, face expressions and orientations. Results show that 24% of times Auric's fails detecting a face. In real world conditions, this percentage may be higher since it depends not only on light con-

ditions, face expressions and orientations, but also depends on how the device is being handled which affects front-facing camera captures that may take picture without a hole face in it.

Discussion

The current face recognition techniques are much less reliable when used in real-world conditions, since they are sensitive to exact conditions in the images, such as light conditions and face orientation. Face preprocessing aims to reduce these problems, such as by making sure the face always appears to have similar brightness and contrast, and perhaps makes sure the features of the face will always be in the same position (such as aligning the eyes and/or nose to certain positions). A good face preprocessing stage will help improve the reliability of the whole face recognition system.

A failure detecting a face and false positives have lower impact on our system, which is an extra log, since it will considered that an intrusion is underway and the interactions will be recorded as well. This means that those failures do not compromise system goals. But false negatives have a greater impact if Auric's is only recording intruder's interactions, which will lead to a lack of logging in the presence of a real intrusion. Yet, if Auric is recording all interactions regardless the intrusion detection result, a failure to detect intruders has low impact, since intruder interactions are recorded as well.

Future Improvements

As future work, we could upgrade face preprocessing stage, which will help improve the reliability of the whole face recognition system[7]. Auric could became more accurate by including a set of new face preprocessing procedure that is supported by a combination of transformations that required face and eye detection. The new procedure has five steps and should will be done after the face detection [7]:

1. **Eye Detection** This process would detect both eyes.

2. **Geometrical transformation and cropping** This process would include scaling, rotating, and translating the images so that the eyes are aligned, followed by the removal of the forehead, chin, ears, and background from the face image.

3. **Separate histogram equalization for left and right sides** This process standardizes the brightness and contrast on both the left and right sides of the face independently.

4. **Smoothing** This process reduces the image noise using a bilateral filter.

5. **Elliptical mask** The elliptical mask process removes some remaining hair and background from the face image.

At the moment, Auric's face preprocessing is only crop the face and standardizes the brightness and contrast on the whole face (histogram equalization), but using this procedure it would became more accurate recognizing trained faces. It is necessary to study the impact on the overall system performance of this preprocessing procedure by comparing the accuracy of the new and the current procedures, and ascertain its viability in terms of performance and accuracy.

Implementation

`recognition` package is responsible for executing face preprocessing, detection and recognition (Figure 4.19).

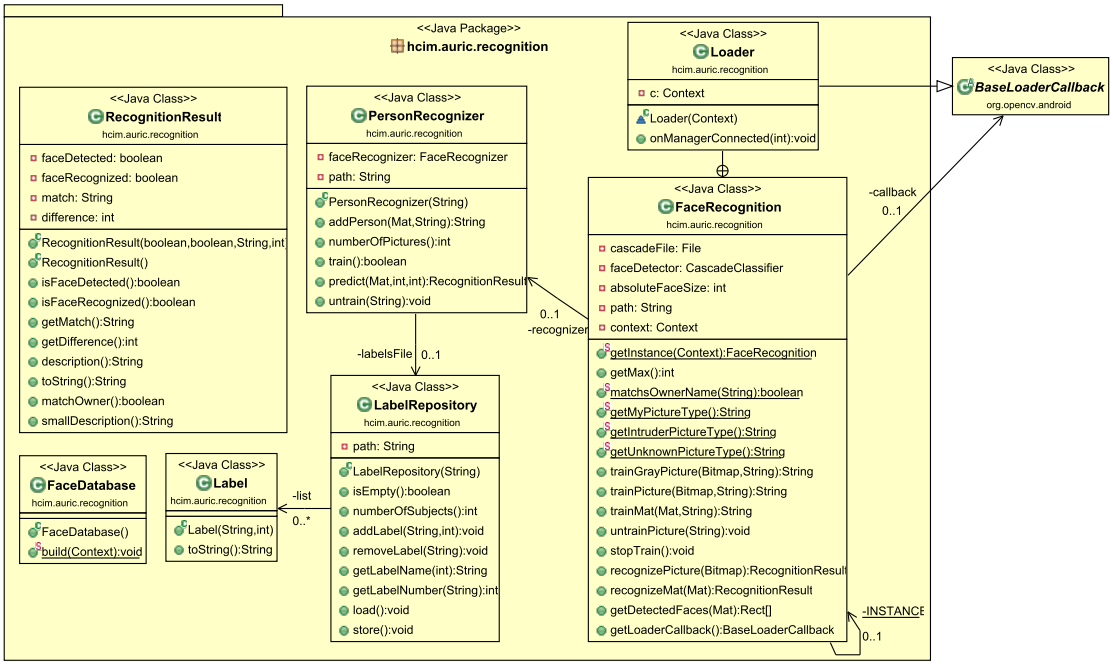
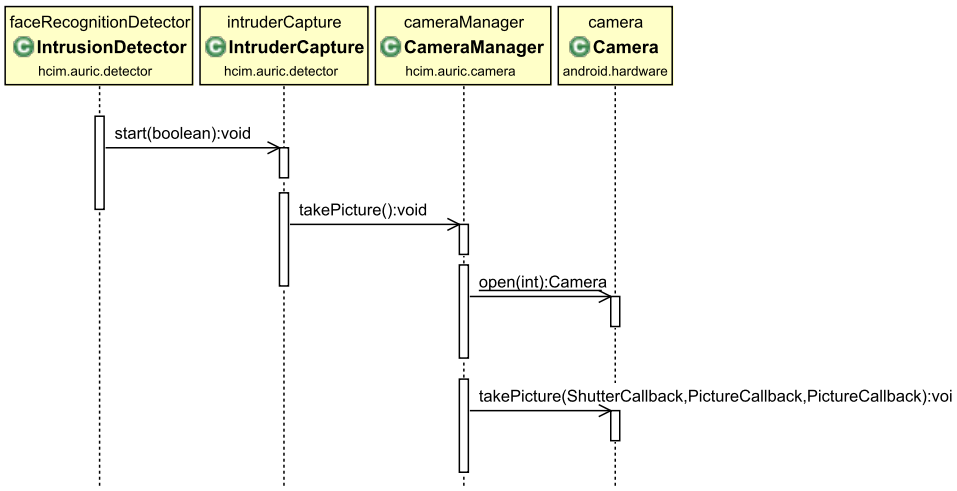


Figure 4.19: `recognition` package overview.

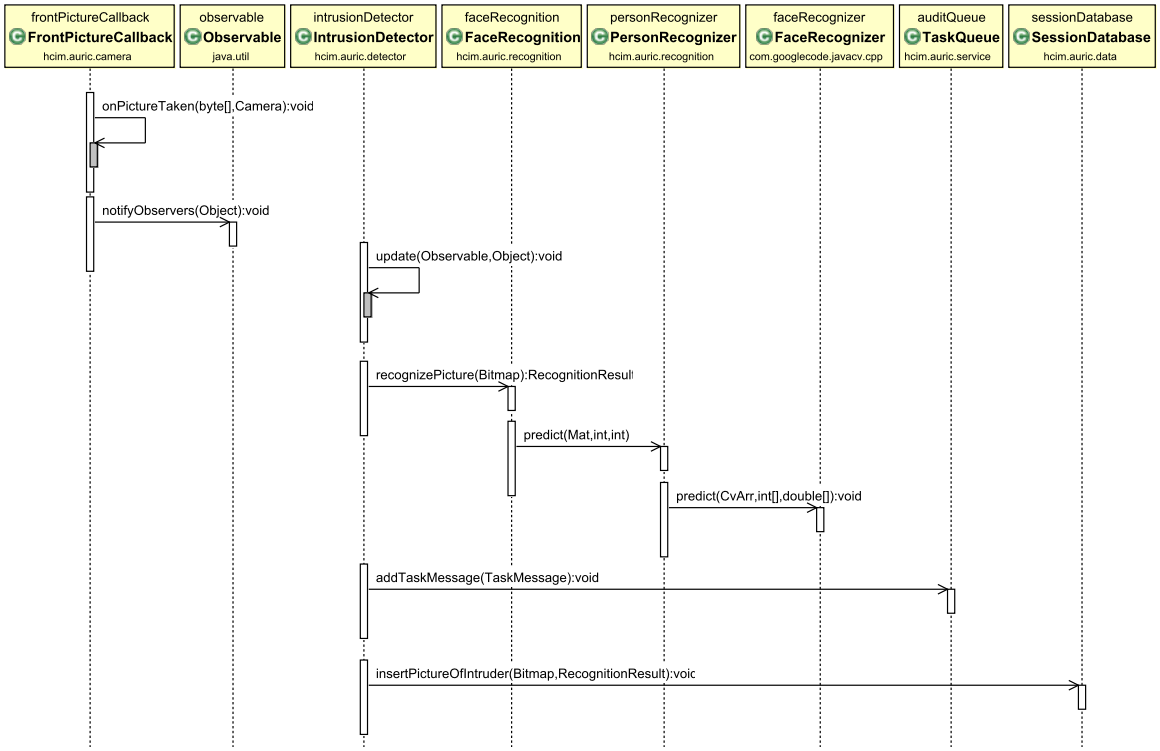
The `FaceRecognition` is the only Java class in this package that communicates with other packages. It offers a set of methods to perform face detection and recognition with different types of inputs, such as Bitmap or matrices. This class encapsulates the different steps of face preprocessing, detection, training and recognition, but is only responsible for face preprocessing and detection, leaving the responsibility of face training and recognition to the `PersonRecognizer` class.

`PersonRecognizer` class has a `LabelRepository` manage labels associated with trained faces, for that reason, contains a list of `Labels`.

Figure 4.20 contains a sequence diagram that illustrates the intrusion detection process, which requires taking a picture (4.20a), process face recognition and report the result(4.20b).



(a) Sequence diagram of taking a picture.



(b) Sequence diagram of receiving the picture taken, processing face recognition, reporting the result and finally, storing the picture taken and the face recognition result.

Figure 4.20: Intrusion detection process.

4.2.8 Data Module

The `data` package is responsible for managing access to Auric's database and shared preferences file (Figure 4.21). The `SessionDatabase` has a `IntrusionTable` which stores `Intrusion` objects, a `IntruderTable` which stores pictures captured using the front-facing camera while the device is being used, and a `SessionTable`

which stores `Session` objects.

The `EventLogDatabase` has a `EventLogTable` which stores `EventBasedLog` objects, i.e. logs produced by the event-based recording method.

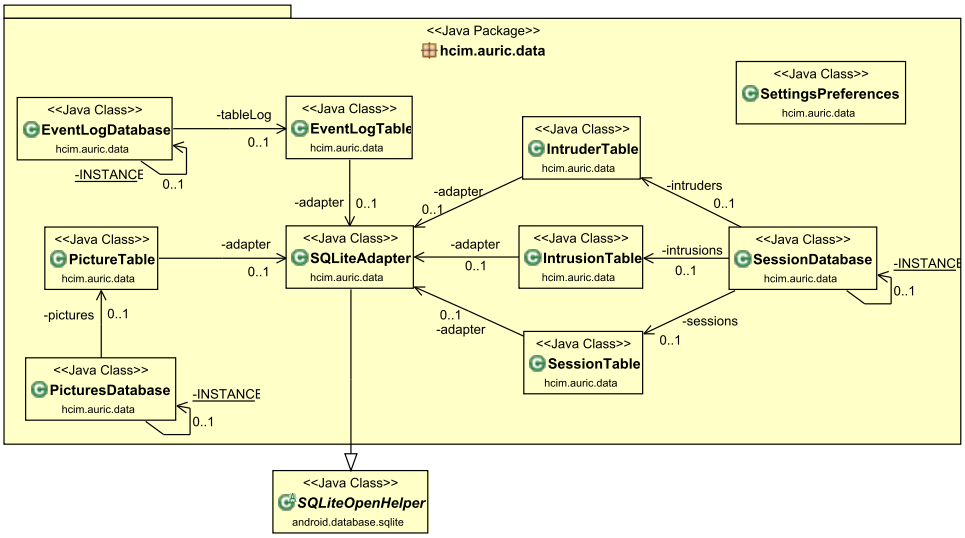


Figure 4.21: `data` package overview.

Auric allows the addition of pictures of other people in order to be recognized as owners. The `PicturesDatabase` has a `PicturesTable` which stores one picture of each trained individual to be recognized as owner. This table is only read by the user interface, to show to the user a picture of each individual trained as device owner. In fact, users do not have or need to know the training procedure, therefore we only store one of each trained picture in this database to show to user which individuals are considered as device owners. Note that the face pictures stored in Android File System alongside with the AT&T face database are already preprocessed (i.e. black and white) and reusing them in the user interface would be little aesthetic.

The `PicturesTable`, `EventLogTable`,`IntrusionTable`, `SessionTable` and `IntruderTable` have an `SQLiteAdapter`, which is an instance of the `SQLiteOpenHelper` abstract class, that manages database creation and access.

4.3 List of Functionalities

Intrusion Detection – Supports a face recognition intrusion detector.

1. The user can choose to hide system notifications.
2. The user can choose how often the system takes pictures and processes a face recognition analysis. Fifteen seconds is the default value.
3. The user can activate or deactivate the Intrusion Detection Service whenever he wants.

4. If the device shuts down while the Intrusion Detection Service was running, when the user turn the device on again the Intrusion Detection Service will restart after boot completed.

Face Recognition

- Enrollment – takes 10 pictures of the device owner and trains them to be recognized as owner.
- The user can define the maximum distance between two pictures that match.
- The user can add pictures of others to be recognized as owners and edit his/her pictures.
- User can test face recognition accuracy.

Recording – Supports two recording methods: screencast and event-based recording. Auric's also supports recording deletion.

1. Event-based recording method
    - **Requires Accessibility Service:** Accessibility Service must be activated manually by the user. Auric redirects the user to accessibility device settings. Also Auric launches a notification if the user forget to activate accessibility service on and by tapping on that notification it will be redirected to accessibility section in device settings.
    - **Visualization** A time-line of applications that were opened is shown, alongside a camera roll of pictures taken with the front-facing camera.
2. Screencast method
    - **Requires root permissions:** This recording option will only appear if the device has root permissions.
    - **Visualization:** A video capture of what happened on the screen is played, with the pictures taken with the front-facing camera rolling in the upper left corner.

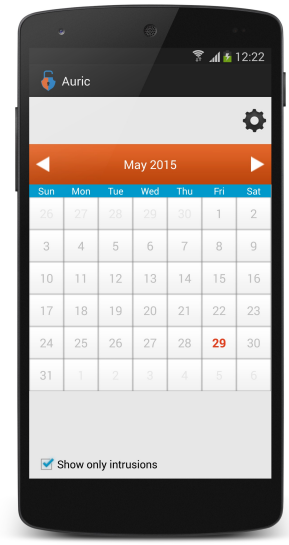
Strategy – Supports two operation strategies

1. **Device Sharing Strategy:** Only records intruder interactions.
2. **Greedy Strategy:** Records all interactions regardless the intrusion detection outcome.

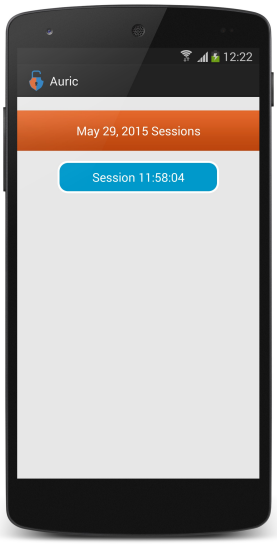
Passcode – the user can set an application access code, in order to prevent an attacker delete recordings or disable it.

4.4 User Interface

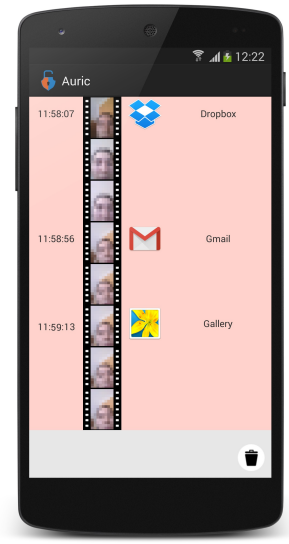
4.4.1 Review recordings



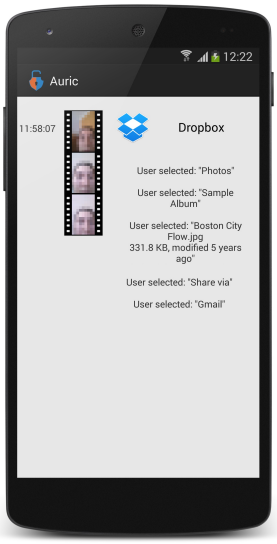
(a) Calendar View. The days when intrusions occurred are marked in red.



(b) List of sessions from a day.



(c) Time-line of applications used in a session, decorated with the pictures taken using the front-facing camera.



(d) Details View: shows user actions within an application.

Figure 4.22: Visualization of logged activity on an Android smartphone using event-based recording method.

We designed an interface that allows the owner to easily access the recordings of intrusions detected (Figure 4.22). We designed a calendar view showing the current month with the current day in bold; days where there were suspected intrusions are selectable and marked in red (Figure 4.22a). By clicking on one of those days, a list of intrusion sessions will appear (Figure 4.22b). Upon selecting a session, the recording of intruder's actions is shown. The way in which sessions are shown depends on recording method selected. If the screen recording method was used, a video capture of what happened on the screen is played, with the pictures taken with the front-facing camera rolling in the upper left corner. If instead the recording was event-based, a time-line of applications that were opened is shown, alongside a camera roll of pictures taken with the front-facing camera (Figure 4.22c). Upon selecting an item on the list, all actions that were performed while using that application are shown (Figure 4.22d). Also the user can zoom intruder's pictures by tapping on them.

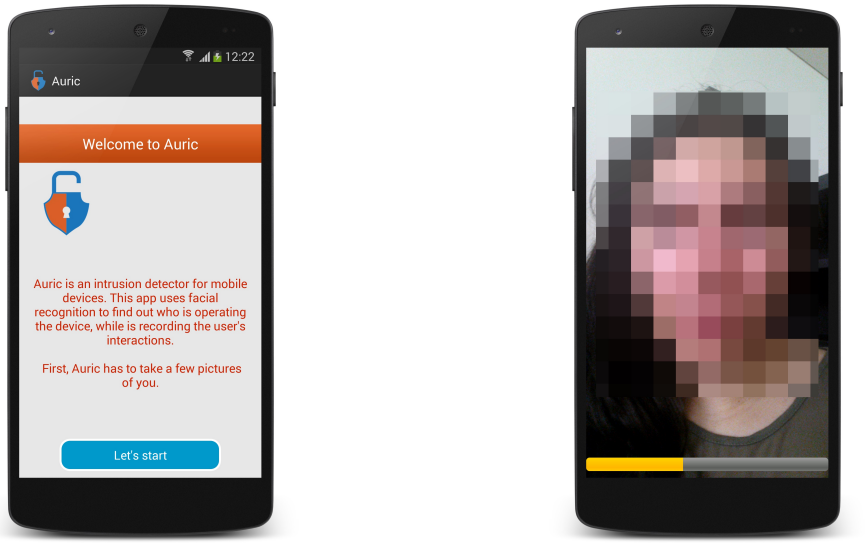
The Figure 4.23 shows the visualization of logged activity on an Android smartphone using screencast recording method, which presents a video of what was showing on the screen and the user's pictures on the top right rectangle.

Figure 4.23: Visualization of an intrusion recorded using screencast method. The top right rectangle shows pictures of the intruders and the background shows captured screenshots.

4.4.2 Enrollment and Settings

The first contact of users with our application is in the enrollment step (Figure 4.24). Firstly, Auric's shows a welcome message that has a brief explanation of its purpose (Figure 4.24a). Secondly, by tapping on "Let's start", it will appear the set up activity.

This activity has a camera preview that captures and trains 10 detected faces of the user. For that reason, the device owner must stay in front of the camera until the yellow bar is full (Figure 4.24b).



(a) Welcome activity.

(b) Set up activity. Inspired by *"Reconocimiento de caras OpenCV"* [2]

Figure 4.24: Welcome and Set up activities.

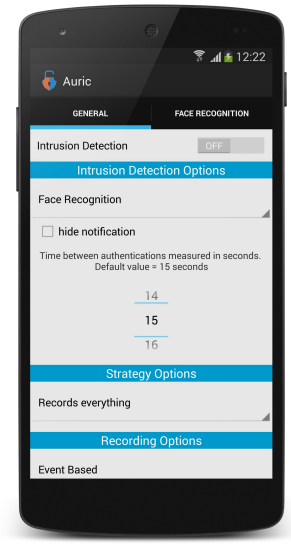
The Figure 4.25 shows the settings activity on an Android smartphone. This activity has two tabs: General and Face Recognition tab. In General Tab, user can activate and deactivate the intrusion detection service, i.e. the monitoring. Also, the user can choose the intrusion detector type, currently it is only available a face recognition intrusion detector. Moreover, the user can choose the operation strategy: record all interactions or record only intruder's interactions (Figure 4.25a). The user can choose the recorder type: screencast or event-based recorder. Furthermore, the user can set a passcode lock to access the application, avoiding deactivation of the service or access logs by a third-party. Finally, the user has an option to delete all registered sessions (Figure 4.25b).

In Face Recognition tab, the user can edit trained pictures, test face recognition accuracy and set the maximum distance between two pictures that match.

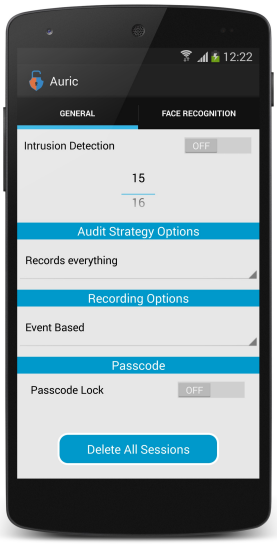
By taping on "Test with Camera" in Face Recognition tab, the user can test face recognition accuracy (Figure 4.26b). This activity has a camera view that processes a face recognition analysis on each frame and displays the results.

By tapping on "Edit Pictures" in Face Recognition tab, the user can view, add and delete his/her pictures. It only show 1 of the 10 pictures taken because the user do not have know the procedure of training, he/she only has to know which individuals are trained to

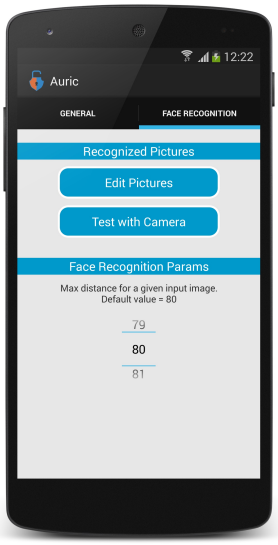
be recognized as device owners (Figure 4.26a).



(a) Settings General Tab.

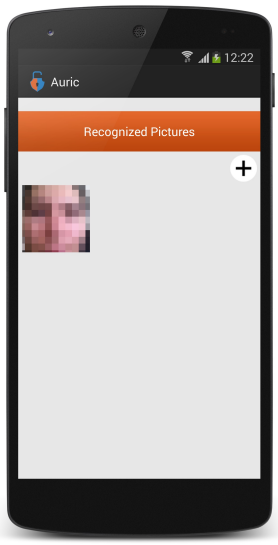


(b) Settings : General Tab scroll down.

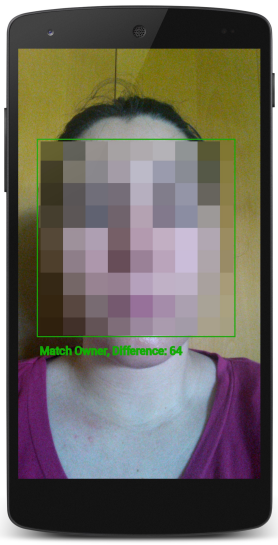


(c) Settings : Face Recognition Tab.

Figure 4.25: General and Face Recognition Tabs from Settings.



(a) Activity for editing pictures known as device owner.

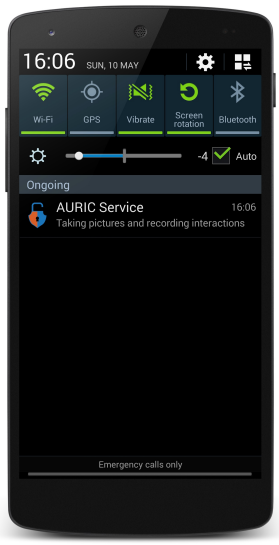


(b) Activity for testing face recognition accuracy. Inspired by *"Reconocimiento de caras OpenCV"*[2]

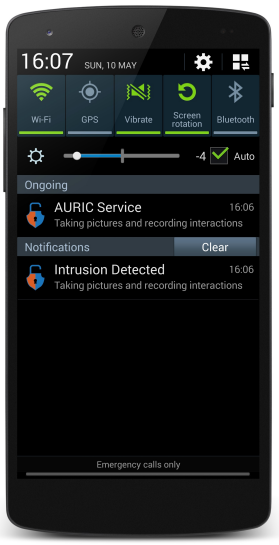
Figure 4.26: Auric's Face Recognition Options.

4.4.3 Running

Auric runs in background and launches two different notifications (Figure 4.27): a permanent and a non-permanent. The permanent notification (Figure 4.27a) ensures that Auric's service will be permanently running until the device owner deliberately stops it. Also it informs the user that Auric is monitoring the device. This notification is launched upon intrusion detection activation on settings activity (Figure 4.25a). The non-permanent notification informs that an intrusion was detected (Figure 4.27b) as a way to deter a potential attack to misbehave.



(a) Permanent notification.



(b) When an intrusion is detect a new notification is launched informing the user of that occurrence.

Figure 4.27: Permanent and intrusion detected notifications launched by Auric.

Chapter 5

Laboratory Study

5.1 Goals

We conducted a laboratory study in order to, firstly, learn about precautions that users have with mobile devices. Secondly, to explore concerns they have about people looking through their data. Thirdly, to explore motivations for device sharing. These topics were essential to contextualize participants in our research and purpose of our application, by eliciting the threat posed by socially-close adversaries.

Finally, we presented our system to participants and in order to understand how they would adopt it. Specifically, we exposed people to Auric simulating two different attacks and then we allowed people to navigate the application and identify what the attacker did and saw. And then try to assess the perceived efficacy of Auric.

In this study, as well as in the next, we opted for a version of the system that only record user interactions using the *event-based recording* method. The *screencast* method requires devices to have root permissions, and hence, we believe that many users would not adopt the system.

5.2 Procedure

We conducted in-depth semi-structured interviews with twelve participants that use mobile devices, such as smartphones and tablets, on a daily basis. The study started with a set of questions about precautions that users have with mobile devices and their concerns over physical attacks. We then introduced our system's purpose and features. Next, we prompted participants to enroll in the system, and conducted a dramatization of two possible situations where our system can be useful. The first was the case where a user leaves its device unattended for a few moments and an attacker takes the chance to snoop through contents. The second was the device sharing case, where the owner hands the phone to someone else to show some content, and that person abuses the situation by accessing other private data.

After the dramatization, where participants acted as the owner, we conducted an exit interview. We started by asking participants for general comments, then probed specifically about our system's usefulness and how they would adopt it, if at all.

5.2.1 Initial interview

The purpose of the initial interview is to explore concerns they have about people looking through their data, to identify defense strategies used by participants, and also to find out want kind of contents, stored in their mobile devices, participants usually share with others. This initial interview also contextualizes participants in our research and purpose of our application. We conducted in-depth semi-structured interviews starting from a set of questions, as follows:

1. How old are you?
2. What is your gender?
3. What is your education level?
4. What is your living situation?
5. What kind of mobile devices do you use daily?
6. Would you say that you lean more towards being a heavy or a light user? (Light: mostly do phone calls and text messages. Heavy: use social networks, games and other mobile applications, multiple times a day)
7. Do you use any security measure against physical intrusion, such as locking?
8. Do you remember any situation that someone else used your device? (Specify situations)
9. Do you consider that your devices to be personal and private?
10. When someone else is using your device, do you feel some sort of worried?

5.2.2 Demonstration

The demonstration of the applications starts with a brief explanation of the application, its purpose and features. Then we prompted participants to enroll in the system, and conducted a dramatization of two possible situations where our system can be useful, as follows:

Use case 1

1. The device is left unattended in a room for a few moments.

2. An attacker unlocks it and checks e-mail and text messages.
3. Before the device owner returns, the attacker leaves it where it was.
4. The device owner checks if someone used his device in his absence.

Use case 2

1. The owner searches for a photo in the device.
2. The owner hands the phone to another person to show that picture.
3. When unsupervised, that person sends it via e-mail.
4. The device is returned to its owner.
5. The device owner checks if the device was only used to view that photo.

5.2.3 Final interview

The purpose of this interview was to perceived how users plan to use this application and also to perceived if it is useful. We conducted in-depth semi-structured interviews starting from a set of questions, as follows:

1. Can you remember any recent past situation in which this application would have been useful?

2. If you could leave this interview with the application installed on your device, would you want?

3. Would you tell someone else that this kind of application is installed on your device?

4. This application has the option to show or not that detected an intrusion and it is recording. What option do you prefer?

5. Do you think that you would catch someone else using your phone without your permission?

6. When an intrusion is detected, this application just records user's pictures and actions, do you think that it could do something else?

7. This application has the option of adding other people's faces, so they won't be considered as intruders. Would you add someone else face?

5.3 Participants

We conducted in-depth semi-structured interviews with twelve participants that use mobile devices, such as smartphones and tablets, on a daily basis. Four were female and eight were male, and ages ranges from 18 to 49 years old ($\mu = 29, \sigma = 8.8$). All participants are from Lisbon, Portugal. Table 5.1 presents data of recruited participants.

Participant	Age	Gender	Education Level	Occupation	Lives with	Mobile Devices	Light or Heavy User	Lock
P1	18	Female	Less than High School	High School Student	Family	Smartphone and tablet	Heavy	Pattern
P2	23	Male	University (Bachelor's)	Student	Family	Smartphone	Light	Pattern
P3	23	Male	University (Bachelor's)	Student	Family	Smartphone	Heavy	PIN
P4	25	Female	University (Bachelor's)	Student	Student Residence	Smartphone	Heavy	PIN
P5	25	Male	University (Bachelor's)	Researcher	Roommate	Smartphone	Heavy	No
P6	25	Male	High School	Student	Family	Smartphone	Heavy	No
P7	28	Male	Graduate School	Researcher	Roommate	Smartphone and tablet	Heavy	PIN
P8	29	Female	Graduate School	Researcher	Family	Smartphone and tablet	Heavy	No
P9	30	Male	Graduate School	Researcher	Family	Smartphone and handheld game console	Light	No
P10	36	Female	Graduate School	Product manager	Family	Smartphones and tablet	Heavy	Smartphones: PIN e Pattern; Tablet: PIN
P11	42	Male	University (Bachelor's)	Product manager	Family	Smartphone and tablet	Heavy	No
P12	49	Male	High School	Product manager	Family	Smartphone and tablet	Heavy	Tablet: PIN; Smartphone: none

Table 5.1: Participants in Laboratory Study.

5.4 Analysis

We recorded audio of the interviews and then transcribed it for analysis. The analysis of the interviews was done using thematic coding inductively[11]. Two researchers (including me) used the first four interview transcripts for code discovery and then independently developed code books. Then, we met and agreed on a preliminary set of codes. Afterwards, we re-coded four interviews and compared the results, and by consensus, agreed on an extended set of codes. Then we coded all the remaining interviews. Reliability was measured in the end and found to be acceptable (Cohen's κ = .92). The reported results are the marginal frequencies found by one of the coders. Table 5.2 presents the set codes used to code all transcribed interviews.

Security Technologies	PIN		User uses Pin to unlock
	Pattern		User uses Pattern to unlock.
Device Sharing	Goal	Text Messages	User shares his device to other person to send text messages
		Camera	User shares his device to other person use the camera.
		Phone Call	User shares his device to other person make a phone call.
		Gaming	User shares his device to other person play games
		Show something	User shares his device to show something
		Internet	User share his device to other person surf the internet
	Defense	It's mine	User doesn't give his device to anyone to show contents, just show them on his hand.
		Supervision	User keeps supervision when someone is using his device.
		Target App Open	User shares his device with the target application already open.
		Keep it closed	User keeps his device around.
		Accounts closed	User takes precautions with the accounts on the device.
Trust			Comments on how trust,affects attitudes towards the use by others.
Adoption			How the user plans to use the technology.
	Deterrence		Comments about using the app in such a way as to inhibit others to misbehave, for instance by informing them that the application is installed or by showing notifications.
	Passive Discovery		Comments about using the app to discover misbehavior by others, but without any explicit intention to change their own behavior in order to catch intruders.
	Entrapment		Same as above, except that user intends to actively create situations where other might be caught misbehaving.
Social Embarrassment			Comments on feeling embarrassed if someone discover that is being recorded.
Anonymity			Comments on how anonymity affects behavior.
Experience of unauthorized use			User shared that someone used his device without authorization or used his device to access unauthorized content.
Suggestions			Only suggestions that were not induced by the interviewer.
	Lock		Add feature that locks the device upon intrusion detection
	Stop Risky Behavior		- same, except only locking when a there is a high risk, e.g. when the intruder does something that is considered sensitive.
	Hide content		Lock or hide contents.
	External notification		Notify external service in case of intrusion, e.g. via e-mail
Want Auric	Yes		User wants Auric installed on his device.
	No		User doesn't want Auric installed on his device.
	Maybe		Not sure.
Inform	Yes		User will inform everybody that he has Auric installed.
	No		User won't inform anyone that he has Auric installed.
	Depends on person		User will inform some people but not others.
	Depends on situation		User will inform others depending on situation.
Show Notification	Yes		User wants to show a notification when an intruder is detected.
	No		User doesn't want to show a notification when an intruder is detected.
	Depends on person		User wants to show a notification when an intruder is detected depending on person that is using the device.
	Depends on situation		User wants to show a notification when an intruder is detected depending on situation.
Catch Someone	Yes		User expects to catch someone using his device without his agreement.
	No		User doesn't expect to catch someone using his device without his agreement.
	Maybe		User thinks that it is possible to catch someone using his device without his agreement.
Add Pictures	Yes		User wants to add a pictures of others.
	No		User doesn't want to add pictures of others.
	Maybe		User thinks that it is possible to add pictures of others.
Private Device	Yes		User considers that his device as private.
	No		User considers that his device as a tool.
Advantages of IDRS	Know Your Enemy		Comments about desire to control social relations; for instance, using Auric to "know who your friends are".
	Damage Control		Comments about capability of controlling damages.
	Additional Security		Comments about

Table 5.2: Set of thematic codes used to coding transcribed laboratory study's interviews.

5.5 Results

5.5.1 Current Usage

All recruited participants, except one, considered their devices to be private, because they contain private or personal data.

A quarter of participants indicated that someone used their devices without authorization or to access unauthorized content.

> "There was a time that I lent the phone for someone else to play, and I ended up discovering that he was not playing. Suddenly I peeked and I saw that person was reading my text messages." (P1)

5.5.2 Device Sharing

All participants reported that they have shared their devices with someone else to perform specific tasks: to show something (10/12), to make phone calls (5/12), to send text messages (4/12), to play games (2/12), to surf the internet (2/12), and for other purposes (3/12), such as using camera or operating the music player.

We identified some defenses used by the participants, including not handing the device to share contents, instead just showing it on their hand (5/12); keeping the mobile device in close proximity at all times (5/12); taking precautions with the accounts on the device, for instance, logging out (4/12), keeping close supervision when someone else is using the device (4/12) and sharing the device with the target application already open (1/12). The large majority of participants (10/12) commented on how trust affects attitudes towards the use by others.

> "I only give the device to someone I trust." (P8)

5.5.3 Adoption

We tried to understand how participants planned to use the technology, if it was available to them. Responses indicate it often depends on the specific situation and on the nature of relationships with others. Half the participants intended to adopt the technology for deterrence. They foresaw using the application in such a way as to inhibit others to misbehave, for instance by informing them that the application was installed or by having it show notifications.

> "To be a deterrent method. Don't touch or I will know." (P12)

Almost all participants (11/12) intended to use our system for passive discovery of intrusions. They suggested using the application to discover misbehavior by others, but without any explicit intention to change their own behavior in order to catch intruders. A

significant portion (4/12), however, indicated that they would use our system for entrapment, intending to actively create situations where others might be caught misbehaving.

> "Maybe I would leave the phone on the table on purpose." (P5)

Only one participant intended to inform everybody that our system was installed. Three indicated that they wouldn't inform anyone. The majority (8/12), however, said they would inform some people but not others. Similarly, when asked if they would set up notifications, only one participant wanted to always show them. The majority (8/12) didn't want to show notifications at all, and 3/12 wanted to show them depending on the situation. Two participants indicated that they wouldn't tell anyone or show notifications because it would be embarrassing if others knew they were being recorded. Four participants commented on how anonymity affects behavior, indicating that if they were to let others know, they might act differently.

> "They do not know what they are doing is being monitored and it is more likely that people will do something, that maybe would not with the user's supervision." (P3)

A quarter of participants expected to catch people using their devices without permission if they were using our application; half did not, and the remaining were unsure. Regarding the possibility of adding pictures of others as authorized users, 7 participants indicated that they would use it, 4 said they wouldn't, and one participant was unsure.

5.5.4 Suggestions

Prompted to give suggestions of ways to react to another person using the device, 3/12 participants suggested locking the device, 2/12 suggested to restrict access to certain contents, 2/12 to lock only if there was a high risk (e.g. when the other user does something that is considered sensitive). Some participants (3/12) also suggested that a notification to an external service could be sent, for instance via e-mail.

5.5.5 Advantages

We asked participants if they saw any advantages of this approach in comparison to the security they already have in place. Two participants indicated that our application would be useful to control damages, by informing what contents were accessed.

> "To anticipate damage, if it is something secret, such as documents. People can have a contingency if they know what happened." (P3)

Half the participants indicated that our system could be used as an additional security measure; for instance, that it could be used along with unlock authentication.

The vast majority of participants, however, saw as the main advantage the ability to regulate social relations, for instance, using our system to "know who your friends are".

> "If I had this application, it would be easy to see who to trust and who could not be trusted" (P10)

> "I think I should just live surrounded by the people that I trust and they would not do this to us [snoop]. This app would help me identifying those people and have control over who you consider as a friend." (P5)

We also asked participants if they would like to leave with our application installed on their devices. Most did (10/12), but still some didn't (2/12). In those cases, technical difficulties were cited, e.g. not having a front-facing camera.

5.6 Discussion

From this study, we conclude that users are interested in the possibility of auditing unauthorized access to contents. Users, the study suggests, could also use this technology to deter people in close social circles from even considering the possibility of snooping through device contents. Most of participants would adopt this technology as a way to regulate social relations, figuring out "who your friends are". In fact, it seems that many participants were more concerned with that than in keeping privacy. From this study, it was visible the deterrence ability of Auric. A participant spontaneously claim that, he would never use another person device because he would fear that Auric was installed. We did not ask this question to the participants, this one just felt free to share that feeling with us. This suggests that Auric could dramatically decrease unauthorized access on mobile devices.

Chapter 6

Field Study

To study how Auric could be adopted and used in real life unconstrained scenarios, we decided to conduct a field study where participants were able to use it for nine days.

6.1 Goals

The main goals of field study were to perceive how participants would actually adopt this system, assess whether the concept is useful and whether it is appropriate to user requirements, and also to detect usability problems. For this study, we installed an instrumented version of our application in the participants' devices. The study lasted nine days, at the end of which, the application was removed. Data was gathered from three meetings with participants.

6.2 Apparatus

The participants used an alternative version of our system that did not take pictures; hence, it did not detect intrusions, and simply recorded all interactions. In the field study version, participants could not review logs on their own till the second day, only during the meetings with the researchers, which had a master password. From the second day on, users could already review the logs.

We chose to prepare this special version, that do not take pictures, because it would be unethical to take pictures of people that did not agree to participate, which would be the case of possible intruders.

6.3 Procedure

The study included three meetings with participants, on the first, second and ninth days. We conducted semi-structured interviews for each meeting, starting from a set of questions, as follows:

Day 1

Step 1: Briefing

Participants were explained system's concept and functionality, the purpose of the study and the procedure, and asked for consent to proceed.

Step 2: Initial Interview

Semi-structured interviews to identify security measures used by participants and also their privacy concerns regarding mobile devices.

1. How old are you?
2. What is your gender?
3. What is your education level?
4. What is your living situation?
5. Do you use any security measure against physical intrusion, such as locking?
6. Do you consider that your devices to be personal and private?
7. Do you always keep your device close or supervised?
8. Do you often share your device with others? (Specify situations)
9. When someone else is using your device, do you feel some sort of worried?
   (a) if not
      i. because they may snoop / they are allowed to snoop
      ii. or because you trust that they will not snoop

Step 3: Installation and Enrollment

Installation of our application in the participant's own device and initial set up. The enrollment was conducted only to detect usability issues.

Day 2

Step 1: Interview

Semi-structured interviews to identify changes in behavior, to assess if participants remembered their interactions, and their expectations regarding unauthorized access.

1. Did the application disturbed the normal function of the device?
2. Can you remember all or almost all interactions with your device?
3. Do you changed your usual usage behavior after installation? (Do you left your phone unattended on purpose to see what happens?)
4. Did you share your device with someone else after our last meet?
5. Do you expect to find any unauthorized access to your device?

Step 2: Sessions Review
The logs were reviewed by participants, and contrasted with their answers in the previous step.

Step 3: Installation
Installation of a new version, with access to logs, and explanation of the differences between versions.

Day 9

Step 1: Comments
Participants were asked to offer general comments about the application concept and experience as a participant.

Step 2: Exit Interview
Semi-structured interviews to summarize how participants adopted the system, and how they changed their behavior or perceptions, if at all.

1. At the moment do you use an unlock authentication mechanism?
2. Did you configured an access code to the application?
3. Did you tell someone that you have this application installed? Why?
4. Did you try to trap someone with this application, to see what happened? Why? How?
5. Did you share your device with someone else after our last meet? (Specify situations)

Step 3: Uninstall
Remove the application from participants devices.

6.4 Participants

We recruited ten Android smartphones users, five were male and five were female, and ages ranges from 22 to 50 years old ($\mu = 35, \sigma = 10.7$). Table 6.1 presents recruited participants data.

Recruitment required that participants had an Android smartphone, with built-in front-facing camera, that use on daily basis, not only for phone calls and text messages, but also to play games, surf the Internet and access social networks. We recruited participants that live in Lisbon, Portugal.

We installed an instrumented version of our application on the participants own devices as stated on the procedure.

Participant	Gender	Age	Education Level	Occupation	Lives with	Android	Smartphone	Locks device
P1	Female	22	University (Bachelor's)	Student	Family	4.1	Samsung Galaxy Nexus S	no
P2	Female	24	University (Bachelor's)	Student	Family	4.4	Wiko Getaway	no
P3	Male	25	High School	Student	Family	4.3	Samsung Galaxy S3 mini	no
P4	Female	29	High School	Student and Office Assistant	Family	4.4	Sony Xperia Z3	no
P5	Male	30	University (Bachelor's)	Security Networks Draftsman	Family	4.4	Samsung Galaxy S2	Android Pattern
P6	Female	37	University (Bachelor's)	Human Resource Technician	Family	4.3	Samsung Galaxy Note 3	PIN
P7	Male	43	University (Bachelor's)	Product Manager	Family	4.3	Samsung Galaxy Core 2	no
P8	Male	46	University (Bachelor's)	Product Manager	Family	4.3	Samsung Galaxy Grand Duos	Android Pattern
P9	Female	48	High School	Banking	Family	4.3	Samsung Galaxy Grand Duos	no
P10	Male	50	High School	Product Manager	Family	4.3	Samsung Galaxy Grand Duos	no

Table 6.1: Participants in Field Study.

6.5 Analysis

We recorded and then transcribed audio of the interviews. The analysis of the interviews was done using thematic coding inductively[11]. We transcribed the interviews and created an initial set of codes. Two researcher's (including me) coded two interviews each, compared the results, and agreed on an extended set of codes. The researchers then re-coded the interviews and measured reliability, which were found to be acceptable (Cohen's κ = .95). I re-coded the remaining interviews. The analysis is based on that researcher's coding. Table 6.2 presents the set codes used to code all transcribed interviews.

Use by third parties	Goal	Text Messages	User shares his device to other person to send text
		Camera	User shares his device to other person use the camera.
		Phone Call	User shares his device to other person make a phone call.
		Gaming	User shares his device to other person play games
		Show something	User shares his device to show something
		Internet	User shares his device to other person surf the Internet
		Others	
	Defense	It's mine	User does not give his device to anyone to show contents, just show them on his hand.
		Supervision	User keeps supervision when someone is using his device.
		Keep it close	User keeps his device around.
		Not Snoop	User trust that friends/family won't snoop
		Nothing to hide	User doesn't feel upset if someone snoop because he has nothing to hide
		Others	others
	Level of concern	Person-dependent	Worries about third-party use depend on type of person / their trust relationship
		Absolute	Concern over ANYONE using the device
		None	No concerns if someone else used the device.
Rethinking			As a result of the intervention (using the app, participating in study) user gained awareness of the threat, and plans to or has acted upon it.
Adoption		Deterrence	Used the app in such a way as to inhibit others to misbehave, for instance by informing them that the application is installed.
		Passive Discovery	Used the app to discover misbehavior by others, but without any changing their own behavior.
		Entrapment	Same as above, except that actively created situations where others could be caught misbehaving.
Experience of unauthorized use			User shared that someone used his device without authorization or used his device to access unauthorized content.
Want Auric		WantAuric.Yes	User wants Auric installed on his device.
		WantAuric.No	User doesn't want Auric installed on his device.
		WantAuric.Unsure	Not sure
BYOD			User brings smartphone or tablet to the workplace and used them as office tools
Usage Experience	Problems	Configuration	User had difficulty setting up the app, including problems with enrollment, or with connecting the accessibility service or runtime service.
		Session	Difficulty in understanding session concept or label.
		Show all sessions	Difficulty in understanding the "show all sessions" check box concept or label.
		Show only intrusions	Difficulty in understanding the "show only intrusion sessions" check box concept or label.
		Battery	Comments about negative impact on battery life.
		Opencv	User offered negative comments related to the need to install OpenCV.
		Log	User had difficulty interpreting the logs collected by the application.
		UI	User wasn't satisfied with the UI's visual appearance
		Doubt	User reported difficulties in determining if the logged activities were their own.
		Boring	User reported that the process of reviewing sessions was boring
		Others	
	Suggestions	Lock	User suggested that the device locks upon intrusion detection
		Delete Options	User perceived the an option to delete objects, e.g. sessions, was missing.
		Pictures	User indicated a preference for showing the pictures taken with the front-facing camera along with the logs.
		Others	others
	Advantages	Secure	User indicated having a stronger sense of security, i.e. feeling more secure with Auric
		Control Children	User indicated that the app covers the use case of controlling a child's activity.
		Passive	User manifested being pleased with having passive security, e.g. that the app doesn't require attention, or that it runs on the background.
		Know who	User pleased that the app allows knowing WHO used the device without permission.
		Know what	User pleased that the app allows knowing WHAT was done in the device by others.
		Others	others

Table 6.2: Set of thematic codes used to coding transcribed field study's interviews.

6.6 Results

6.6.1 Use by third-parties

We again examined what motivated participants to share their mobile devices, their defenses to protect personal data and level of concern about a third-party using their device.

Goals

All participants reported that they have shared their devices with someone else to perform specific tasks: to show something (7/10) such as a photographs, to make a phone call (5/10), to play games (4/10), to send a text message (4/10), to surf the Internet (3/10) and others (2/10).

Defenses

Reported defenses included not handing over the device when showing contents (3/10); keeping supervision when someone is using their device (2/10); keeping the device around (8/10), trusting that friends and family will not snoop through device contents (6/10); and not storing very sensitive information on devices (4/10).

Level of concern

Regarding worries about having someone looking through their device data, 5/10 participants reported that it depended on the type of person or their trust relationship; and 2/10 that they were concerned over anyone using their devices. Only 1 participant wasn't concerned at all.

6.6.2 Bring your own device

Two participants reported working in organizations that adopted BYOD policies. They use their own devices to handle professional e-mail and documents. Hence, their concerns were not only over personal data, but also sensitive work products.

> "This [device] has information about the two parts of my life, my personal and professional life." (P8)

6.6.3 Experiences of unauthorized use

Two participants shared that someone used their devices without authorization, prior to the study. P9 reported a suspicion that a colleague snooped through her device after sharing it for a phone call. P6 reported that her tablet was accessed without permission to consult specific data.

6.6.4 Impact of participation

As a result of participating in study, two participants reported an increased awareness of the threat, and plans to act on it. Before the field study, P9 did not use unlock authentication because it was inconvenient, but afterwards decided to set it up. P2 said that this study helped her realize the sensitive data stored on her device, and that she would now set unlock authentication also.

6.6.5 Usage experience

Adoption

Only one participant reported adopting our system as a deterrent. P9 indicated that she informed her family that this kind of application was installed on her device in a way to discourage them to use her phone. Eight participants used the application to discover misbehavior by others, but without any changing their own behavior. One participant used the application for entrapment, we.e. actively creating situations where others could be caught misbehaving.

Problems

All but one participant had some kind of difficulty setting up the application, including problems with enrollment, or with connecting the accessibility service or runtime service. These were usability problems that can be easily overcome by creating a wizard to help users through the initial configurations steps.

Half the participants expressed concerns over negative impact on battery life.

Seven participants had some kind of difficulty in interpreting the logs collected by the application. Some participants initially didn't understand the meaning of some the applications that appeared on the logs, such as home, lock screen, and launcher, because these packages are not commonly seen as being apps. This usability problem was mitigated early, and users received an update where packages related with system activities were filtered out.

Four participants reported difficulties in determining if the logged activities were their own. This problem was expected in the instrumented version used in the field study, which doesn't capture pictures because of ethics concerns. Indeed, 9/10 participants reported that they would prefer seeing pictures taken with the front-facing camera along with the logs.

90% of participants expressed difficulty in understanding the "show all sessions" check box concept or label. The "show all sessions" check box was to filter sessions, when active shows all recorded sessions otherwise will only show intrusion sessions, i.e. sessions performed by a third-party. Due to this difficulty, the "show all sessions" check

box was changed to “show only intrusion sessions” check box. In this case, when it is enabled, it will show only intrusion sessions; otherwise, it will show all sessions. Only one participant express difficulty in understanding the “show only intrusion sessions” check box concept or label.

Four participants expressed difficulty in understanding some labels in the app, such as “session”, which we meant as the period of time between the device waking up and being again turned off. The main reason why this happened was because users do not represent their usage as a set of sessions, but as continuous. This issue warrants further evaluation of design alternatives, for instance presenting a condensed time-line, with expandable logs for whole days.

Advantages

Four participants indicated having a stronger sense of security with our application.

> “I feel more secure because if someone uses it we will know.” (P4)
> “The icon makes me feel more secure, since everybody can see that is being recorded.” (P7)

Three participants indicated that the application could be used to monitor a child’s activity.

> “I could use it to know what my son does on his tablet” (P6)

The majority of participants (8/10) manifested being pleased with having passive security, e.g. that the app does not require attention, or that it runs on the background.

> “I think it is a type of application that does not require attention. It is running and when you feel the need you see the recordings. I think that’s positive.” (P5)

> “It doesn’t ask me for a PIN or a sketch to use my phone.”(P7)

Eight participants were pleased that the application allows them to know *who* used the device without permission, thus confirming our observation in the first study, that users want to closely regulate their trust relationships.

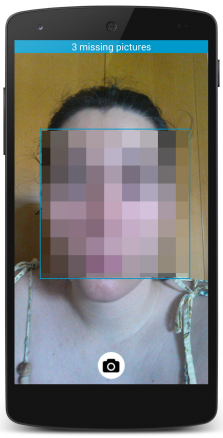
Four participants were pleased that the application allows knowing *what* was done in the device by others, which indicates a type of adoption that focuses more on damage control, as do traditional intrusion detection systems.

6.7 Discussion

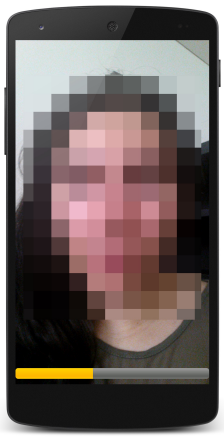
In this study, we found that users are indeed concerned about third-parties looking through their mobile device data, and that those concerns are often dependent on trust relationships. Despite that, a large number of participants did not use unlock authentication because it is inconvenient or tedious. Our approach allows users to protect their mobile

device contents without having to expend additional effort. Also, our system offers users the opportunity to know if their phone was attacked, to know who was the attacker, and what they did.

The impact on battery consumption demands improvements, since 50% of participants expressed concerns about it. Also there are some usability issues about set up that must be solved. After this study we updated the initial picture set up. We designed an Android activity that captures pictures of the device owner automatically without requiring pressing a button to capture. Figure 6.1 presents the old (Figure 6.1a) and new (Figure 6.1b) versions of the enrollment.



(a) Set up activity used in Field Study. Some users did not understand that they have to press the capture button to conduct the set up.



(b) Updated set up. Captures user's facial pictures automatically, without requiring any explicit interaction.

Figure 6.1: Old vs new set up activities.

Chapter 7

Conclusion

Auric is a security and privacy application for Android smartphones. It is an Intrusion Detection and Response System against unauthorized access and abusive usage by third-parties. Auric is capable of dissuading social-close adversaries from accessing private content. This application is a starting point for the development of security applications for mobile devices more appropriate to short and frequent usage.

The advantages of Auric is being transparent and effortless. The transparent operation of Auric is one of the best characteristics. Despite the permanent notification that informs the operator that his actions are being recorded and his pictures are being taken, users cannot perceive that a tool like this is running on the device.

Auric is an effortless security measure because the user does not have to change the regular use of the device. The conventional unlock authentication requires an additional step which is entering a PIN, password, pattern or waiting for a biometric analysis, and without overcoming this step, the user cannot have access to the device contents. These authentication mechanisms represent a lot of effort to the users since interactions with mobile devices are short and frequent. Therefore, Auric is more appropriate than unlock authentication in terms of effort for the user.

One of the major advantages of Auric is being a complement tool intended to dissuade socially-close adversaries from snooping through device contents. This system is not a replacement of unlock authentication, since unlock authentication mechanisms are useful when a device is lost or stolen.

Our approach offers intrusion detection and response capabilities to end-users, specifically for the risk of physical intrusion by socially-close adversaries. Our system can act as a deterrent and also as a tool for incident management.

In two user studies, we found that users are indeed concerned about third-parties looking through their mobile device data, and that those concerns are often dependent on trust relationships. Despite that, a large number of participants did not use unlock authentication because it is inconvenient or tedious. Our approach bridges this gap, achieving users' desire to protect the contents of their mobile devices, without having to expend additional

effort. Our system also offers to users the opportunity to know if their phone was attacked, to know who was the attacker, and what they did.

Intrusion detection will be the future of security in mobile devices against physical intrusions. Since building a perfect security system is very difficult or even impossible, so the only solution is detecting and reacting to intrusions, i.e. detecting usage by third-parties, and reacting upon detection.

7.1 Limitations

7.1.1 Face Recognition

One known limitation of face recognition is that it's accuracy strongly depends on light conditions, camera quality and framing of the pictures. In this system, the lack of accuracy can somewhat be mitigated by the fact that multiple pictures are taken. Furthermore, since a false positive only produces an additional log, the recognition algorithm can be optimized to minimize false negatives. Since the objective of our studies was to assess the feasibility of mobile intrusion detection systems, and how they would be adopted, we leave further investigation into specific biometric techniques (face or otherwise) for future work.

7.1.2 Privacy Implications

As much as device owners see the potential in our application to protect their privacy, their friends and family might see it as infringing on their own privacy. In fact, our software could be used as an offensive tool, and the owner might seek to share the device in order to see what other people do, which may include accessing their own accounts. The fact that 3 participants in the field study suggested that our system could be used to control children's activities is revealing of the possibility of misuse. However, we find that if someone's intent were to spy on others, there are several tools better suited for the job than our software, and there's nothing to stop someone from installing spyware on their own devices.

The developed intrusion detection system authenticates the user through face recognition, which means it has to capture a picture of his face and process a face recognition analysis.

Scenario 1 - Device sharing to make a phone call *James has Auric installed on his smartphone to prevent friends and family to snoop through its contents. Peter is a friend of James and is hanging out with him. Peter forgot his smartphone at home. Peter asks James to make a phone call from his smartphone. James lends his device to Peter. Auric*

detects an intrusion and starts recording Peter's interactions. Peter makes the phone call and returns the phone to James. James can know who Peter called.

Scenario 2 – Device sharing to check Facebook
James has Auric installed on his smartphone to prevent friends and family to snoop through its contents. Mary is a friend of James and is hanging out with him. Mary forgot her smartphone at home. Then she asks James if she could check her Facebook messages on his smartphone. James logs out his Facebook account and lends his device to Mary. Auric detects an intrusion and starts recording Mary's interactions. She logs in to her Facebook account and checks messages. Then Mary logs out her account and returns the phone to James. James can read the Facebook messages that Mary read.

In the second scenario, if James is using event-based recording method, Mary passwords remain confidential, because this method discards events related with passwords. But if James was using screen recorder, he may able to induce the password by watching the video.

Third-parties must know that their actions are being recorded in order to protect their data. For that reason, Auric shows a notification, not only to deter unauthorized accesses to contents but also to inform users that their actions are being recorded and their pictures are being taken. Otherwise it would be unethical and could cause several private data protection issues.

7.1.3 Performance

Because we wanted to examine feasibility and adoption issues, we did not conduct a formal performance analysis, nor did we optimize the software after the field study, leaving that for future work. Given the possible impacts on battery and data storage, we conducted only a test in a heavy user's device during 8h, between 5pm and 1am. The device was a Wiko Getaway smartphone with Android 4.4. Our system was configured to take pictures every fifteen seconds, while the device was being used, and to record all interactions regardless of the recognition result. In that period, our system was responsible for 5% of battery consumption, and occupied 140MB of storage. The space occupied by the application is essentially due to the size and number of photographs taken during monitoring. The impact on storage is significant and it could be improved by compressing pictures and/or offloading them to the cloud. The impact of battery consumption was not significant but could still be improved.

7.2 Future Work

Future work on our implementation will target the usability issues that were identified, on improving performance, and on improving the accuracy of facial recognition by implementing an algorithm based on face and eye detection.

7.2.1 Understanding the impact of unauthorized accesses

It would be very interesting to study users' motivations to use someone else personal mobile devices without permission and also understand the impact of the snooping behavior on the victim. In the Laboratory Study (Chapter 5), a participant reported a suspicion of an unauthorized access by a co-worker that led to a negotiation failure. The same source reported that this kind of situation are usual in business environment. It is very important to differentiate the family/friends environments from work/business environments. It would be interesting to further study security and privacy issues of BYOD regarding personal and business data. Also, further study the impact of device sharing in business environments. And finally, study how Auric would be used in business environments, how useful it would be and what kind of features are desirable for this environment.

7.2.2 New Intrusion Detection

Auric detects an intrusion by identifying through face recognition if the device is being used by a third-party. Our system is prepared to accommodate new intrusion detection methods besides face recognition. Specially if they are based on biometric characteristics, because those methods are appropriate for a transparent continuous authentication. There are a lot of proposals on authentication based on biometrics that could also be supported by Auric, such as touch analysis and keystroke dynamics. It would be interesting to implement and to compare not only the efficiency of each method, but also the accuracy.

Intrusion Detection Alarm

Intrusion Detection Systems work as burglar alarm, they activate an alarm upon violation detection. This alarm can be audible (producing noise), visual (producing lights), or they can be silent (sending an e-mail message). Auric's alarm is a notification that informs the user that an intrusion occurred. But as an intrusion detection system, Auric could support others alarms. For instance, Auric could send an e-mail informing the device owner that someone else used the device and also providing the intrusion's log. A more deterrent approach would be to launch a pop-up window informing the third-party that the device is being monitored, without blocking it. This approach enables device sharing, but the attacker is explicitly informed that his/her actions are being recorded. With almost all

IDSs, system administrators can choose the configuration of the various alerts associated with each security level. Auric could also support this feature.

7.2.3 New operation strategies

Auric could have a strategy that depends on task sensitivity. In other words, Auric could only record third-party's interactions, if his/her actions are considered sensitive. For instance, if the third-party is playing a game, Auric does not record interactions, since it has little impact on device owner's privacy. But if the third-party starts reading text messages or e-mails, Auric starts recording user's interactions.

7.2.4 New reactions to intrusion

Auric is able to detect and react to intrusions. Since it is possible to detect an intrusion and react by recording intruders actions. Auric could also react in other ways, such as lock device or just lock specific tasks. In this case, Auric is able to limit the loss from an intrusion. Specifically, the system could lock the device when it is being used by someone else rather than the owner or lock in case the attacker access a specific application. This assumes that the user can create a black list of applications that no one can have access.

7.2.5 Usability

We have detected some usability issues during field study. All but one participant had some kind of difficulty setting up the application, including problems with enrollment, connecting the accessibility service or runtime service. These were usability problems that can be easily overcome by creating a wizard to help users through the initial configurations steps.

Four participants expressed difficulty in understanding some labels in the app, such as "session", which we meant as the period of time between the device waking up and being again turned off. The main reason why this happened was because users do not represent their usage as a set of sessions, but as continuous. This issue warrants further evaluation of design alternatives, for instance presenting a condensed time-line, with expandable logs for whole days.

7.3 Others appliances

The event-based recording method could be used to other purposes, such as to study the mobile devices usage. Specifically, a study could be conducted in order to perceive how long and what kind and of applications are used during a large period of time.

Three participants of the field study indicated that the application would be useful to control their child's activity on mobile devices: to check which websites they consult, to

whom they are talking to and others activities. This way they could have better control of their child's activity on mobile devices, since there are several problems related with children and Internet.

Bibliography

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer vision-eccv 2004*, pages 469–481. Springer, 2004.

[2] Robotic Apps. Reconocimiento de caras OpenCV – Android Apps on Google Play. Google Play Store, August 2013.

[3] Eyal Arubas. Face Detection and Recognition (Theory and Practice). `http://eyalarubas.com/face-detection-and-recognition.html`, April 2013.

[4] Astorre. Gotcha! – Android Apps on Google Play. `https://play.google.com/store/apps/details?id=astorre.gotcha`, 2013.

[5] Samuel Audet. JavaCV – Java interface to OpenCV and more. `https://code.google.com/p/javacv/`.

[6] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge Attacks on Smartphone Touch Screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.

[7] D. L. Baggio, S. Emami, D. M. Escriva, K. Ievgen, N. Mahmood, J. Saragih, and R. Shilkrot. *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing, Limited, 2012.

[8] Peter N. Belhumeur, João P Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[9] Noam Ben-Asher, Niklas Kirschnick, Hanul Sieger, Joachim Meyer, Asaf Ben-Oved, and Sebastian Möller. On the need for different security methods on mobile phones. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 465–473, 2011.

[10] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[11] Kathy Charmaz. *Constructing grounded theory*. Sage, 2014.

[12] Erika Chin, Adrienne Porter Felt, Vyas Sekar, and David Wagner. Measuring User Confidence in Smartphone Security and Privacy. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 1:1–1:16, New York, NY, USA, 2012. ACM.

[13] Nathan Clarke, Sevasti Karatzouni, and Steven Furnell. Flexible and transparent user authentication for mobile devices. In *Emerging Challenges for Security, Privacy and Trust*, pages 1–12. Springer, 2009.

[14] Ronald V. Clarke. *Situational crime prevention*. Criminal Justice Press Monsey, NY, 1997.

[15] Google Code. Android Screenshot Library | Library for taking screenshots on Android platform. `https://code.google.com/p/android-screenshot-library/`, 2011.

[16] Heather Crawford, Karen Renaud, and Tim Storer. A framework for continuous, transparent mobile device authentication. *Computers & Security*, 39, Part B(0):127 – 136, 2013.

[17] Alexander De Luca, Marian Harbach, Emanuel von Zezschwitz, Max-Emanuel Maurer, Bernhard Ewald Slawik, Heinrich Hussmann, and Matthew Smith. Now you see me, now you don't: Protecting smartphone authentication from shoulder surfers. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2937–2946, New York, NY, USA, 2014. ACM.

[18] Dorothy E. Denning. An intrusion-detection model. *IEEE Trans. Softw. Eng.*, 13(2):222–232, February 1987.

[19] Roberto Di Pietro and Luigi V Mancini. *Intrusion detection systems*, volume 38. Springer Science & Business Media, 2008.

[20] Serge Egelman, Sakshi Jain, Rebecca S. Portnoff, Kerwell Liao, Sunny Consolvo, and David Wagner. Are You Ready to Lock? Understanding User Motivations for Smartphone Locking Behaviors. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 750–761, New York, NY, USA, 2014. ACM.

[21] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. A Survey of Mobile Malware in the Wild. In *Proceedings of the 1st ACM Workshop*

on Security and Privacy in Smartphones and Mobile Devices, SPSM '11, pages 3–14, New York, NY, USA, 2011. ACM.

[22] Lorenzo Gomez, Iulian Neamtiu, Tanzirul Azim, and Todd Millstein. RERAN: Timing- and touch-sensitive record and replay for Android. In *2013 35th International Conference on Software Engineering (ICSE)*. IEEE, May 2013.

[23] Abdenour Hadid, Matti Pietikäinen, and Timo Ahonen. A discriminative feature space for detecting and recognizing faces. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–797. IEEE, 2004.

[24] Alina Hang, Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. Too Much Information!: User Attitudes Towards Smartphone Sharing. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, pages 284–287, New York, NY, USA, 2012. ACM.

[25] Alina Hang, Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. FaceProfiles: Inconspicuous, Private and Secure Mobile Device Sharing . In *Workshop on Inconspicuous Interaction at CHI 2014*, Toronto, Canada, 2014.

[26] Marian Harbach, Emanuel von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 213–230, Menlo Park, CA, July 2014. USENIX Association.

[27] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. CASA: Context-aware Scalable Authentication. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 3:1–3:10, New York, NY, USA, 2013. ACM.

[28] Eiji Hayashi, Oriana Riva, Karin Strauss, A. J. Bernheim Brush, and Stuart Schechter. Goldilocks and the Two Mobile Devices: Going Beyond All-or-nothing Access to a Device's Applications. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 2:1–2:11, New York, NY, USA, July 2012. ACM.

[29] IBM. IBM BYOD – Bring Your Own Device – United States. `http://www.ibm.com/mobilefirst/us/en/bring-your-own-device/byod.html`, 2015. [Online; accessed 27-April-2015].

[30] Google Inc. Saving Data | Android Developers. http://developer.android.com/training/basics/data-storage/index.html. [Online; accessed 22-April-2015].

[31] Google Inc. Saving Data in SQL Databases | Android Developers. http://developer.android.com/training/basics/data-storage/databases.html. [Online; accessed 22-April-2015].

[32] Google Inc. Saving Key-Value Sets | Android Developers. http://developer.android.com/training/basics/data-storage/shared-preferences.html. [Online; accessed 22-April-2015].

[33] Google Inc. Android Debug Bridge | Android Developers. http://developer.android.com/tools/help/adb.html, 2014.

[34] Google Inc. Controlling the Camera | Android Developers. http://developer.android.com/training/camera/cameradirect.html, 2014.

[35] Google Inc. Service | Android Developers. http://developer.android.com/reference/android/app/Service.html, 2015. [Online; accessed 5-May-2015].

[36] Anil K. Jain and Stan Z. Li. *Handbook of Face Recognition*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[37] Markus Jakobsson, Elaine Shi, Philippe Golle, and Richard Chow. Implicit Authentication for Mobile Devices. In *Proceedings of the 4th USENIX Conference on Hot Topics in Security*, HotSec'09, pages 9–9, Berkeley, CA, USA, August 2009. USENIX Association.

[38] Amy K. Karlson, A.J. Bernheim Brush, and Stuart Schechter. Can I Borrow Your Phone? Understanding Concerns When Sharing Mobile Phones. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI '09*, New York, New York, USA, April 2009. ACM Press.

[39] HG Kayacık, M Just, L Baillie, D Aspinall, and N Micallef. Data Driven Authentication: On the Effectiveness of User Behaviour Modelling with Mobile Device Sensors. In *Proceedings of IEEE Security & Privacy Workshop on Mobile Security Technologies (MoST)*, 2014.

[40] DoMobile Lab. AppLock – Android Apps on Google Play. https://play.google.com/store/apps/details?id=com.domobile.applock, 2014.

[41] Lingjun Li, Xinxin Zhao, and Guoliang Xue. Unobservable Re-authentication for Smartphones. In *NDSS*, 2013.

[42] Yunxin Liu, Ahmad Rahmati, Yuanhe Huang, Hyukjae Jang, Lin Zhong, Yongguang Zhang, and Shensheng Zhang. xShare: Supporting Impromptu Sharing of Mobile Phones. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 15–28, New York, NY, USA, June 2009. ACM.

[43] Invisibility Ltd. Recordable – Android Apps on Google Play. `http://recordable.mobi/`, 2014.

[44] Diogo Marques, Tiago Guerreiro, and Luis Carriço. Measuring snooping behavior with surveys: It's how you ask it. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 2479–2484, New York, NY, USA, 2014. ACM.

[45] Diogo Marques, Tiago Guerreiro, Luís Duarte, and Luís Carriço. Under the table: Tap authentication for smartphones. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, BCS-HCI '13, pages 33:1–33:6, Swinton, UK, UK, 2013. British Computer Society.

[46] Muhammad Muaaz. A Transparent and Continuous Biometric Authentication Framework for User Friendly Secure Mobile Environments. In *Adjunct Proceedings the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013)*, 2013.

[47] Ildar Muslukhov. Survey: Data Protection in Smartphones Against Physical Threats. 2010.

[48] Ildar Muslukhov, Yazan Boshmaf, Cynthia Kuo, Jonathan Lester, and Konstantin Beznosov. Understanding users' requirements for data protection in smartphones. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering Workshops*, ICDEW '12, pages 228–235, Washington, DC, USA, 2012. IEEE Computer Society.

[49] Ildar Muslukhov, Yazan Boshmaf, Cynthia Kuo, Jonathan Lester, and Konstantin Beznosov. Know Your Enemy: The Risk of Unauthorized Access in Smartphones by Insiders. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 271–280, New York, NY, USA, August 2013. ACM.

[50] OpenCV. Face Recognition with OpenCV. `http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html`, 2015. [Online; accessed 5-May-2015].

[51] Stack Overflow. What is the advantage of Using SQLite rather than File? | Stack Overflow. `http://stackoverflow.com/questions/19946298/what-is-the-advantage-of-using-sqlite-rather-than-file`. [Online; accessed 22-April-2015].

[52] Oriana Riva, Chuan Qin, Karin Strauss, and Dimitrios Lymberopoulos. Progressive Authentication: Deciding When to Authenticate on Mobile Phones. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 301–316, Bellevue, WA, 2012. USENIX.

[53] André Rodrigues. Mobile system-wide assistive technology. Master's thesis, Faculty of Science of the University of Lisbon, 2014.

[54] Ferdinando S. Samaria and Andy C. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE, 1994.

[55] sqlite.org. Internal Versus External BLOBs in SQLite. `https://www.sqlite.org/intern-v-extern-blob.html`. [Online; accessed 22-April-2015].

[56] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[57] Emanuel von Zezschwitz, Paul Dunphy, and Alexander De Luca. Patterns in the wild: A field study of the usability of pattern and pin-based authentication on mobile devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 261–270, New York, NY, USA, 2013. ACM.

[58] Emanuel Von Zezschwitz and Alina Hang. Towards Privacy-Aware Mobile Device Sharing. In *4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use (IWSSI/SPMU)*, 2012.

[59] Cheng-Yao Wang, Wei-Chen Chu, Hou-Ren Chen, Chun-Yen Hsu, and Mike Y. Chen. EverTutor: automatically creating interactive guided tutorials on smartphones by user demonstration. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, New York, New York, USA, April 2014. ACM Press.

[60] Michael E. Whitman and Herbert J. Mattord. *Principles of Information Security*. Course Technology Press, Boston, MA, United States, 4th edition, 2011.