UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

BUILDING AND EVALUATING AN INCONSPICUOUS SMARTPHONE AUTHENTICATION METHOD

Diogo Homem Marques

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

2013

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

BUILDING AND EVALUATING AN INCONSPICUOUS SMARTPHONE AUTHENTICATION METHOD

Diogo Homem Marques

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

Trabalho orientado pelo Prof. Doutor Luís Manuel Pinto da Rocha Afonso Carriço

2013

Abstract

As our intimate lives become more tangled with the smartphones we carry, privacy has become an increasing concern. A widely available option to mitigate security risks is to set a device so that it locks after a period of inactivity, requiring users to authenticate for subsequent use.

Current methods for establishing one's identity are known to be susceptible to even rudimentary observation attacks. The mobile context in which interactions with smartphones are prone to occur further facilitates shoulder-surfing.

We submit that smartphone authentication methods can be better adapted to the mobile context. Namely, the ability to interact with the device in an inconspicuous manner could offer users more control and the ability to self-protect against observation.

Tapping is a communication modality between a user and a device that can be appropriated for that purpose. This work presents a technique for employing sequences of taps, or *tap phrases*, as authentication codes. An efficient and accurate tap phrase recognizer, that does not require training, is presented.

Three user studies were conducted to compare this approach to the current leading methods. Results indicate that the tapping method remains usable even under inconspicuous authentications scenarios. Furthermore, we found that it is appropriate for blind users, to whom usability barriers and security risks are of special concern.

Keywords: security, usability, mobile, gestures, tapping

Resumo

Os smartphones que trazemos connosco estão cada vez mais entranhados nas nossas vidas íntimas. Estes dispositivos possibilitam novas formas de trabalhar, de socializar, e até de nos divertirmos. No entanto, também criaram novos riscos à nossa privacidade.

Uma forma comum de mitigar estes riscos é configurar o dispositivo para bloquear após um período de inatividade. Para voltar a utilizá-lo, é então necessário superar uma barreira de autenticação. Desta forma, se o aparelho cair das mãos de outra pessoa, esta não poderá utilizá-lo de forma a que tal constitua uma ameaça.

O desbloqueio com autenticação é, assim, o mecanismo que comummente guarda a privacidade dos utilizadores de smartphones. Porém, os métodos de autenticação atualmente utilizados são maioritariamente um legado dos computadores de mesa. As palavras-passe e códigos de identificação pessoal são tornados menos seguros pelo facto de as pessoas criarem mecanismos para os memorizarem mais facilmente. Além disso, introduzir estes códigos é inconveniente, especialmente no contexto móvel, em que as interações tendem a ser curtas e a necessidade de autenticação atrapalha a prossecução de outras tarefas.

Recentemente, os smartphones Android passaram a oferecer outro método de autenticação, que ganhou um grau de adoção assinalável. Neste método, o código secreto do utilizador é uma sucessão de traços desenhados sobre uma grelha de 3 por 3 pontos apresentada no ecrã táctil.

Contudo, quer os códigos textuais/numéricos, quer os padrões Android, são suscetíveis a ataques rudimentares.

Em ambos os casos, o canal de entrada é o toque no ecrã táctil; e o canal de saída é o visual. Tal permite que outras pessoas possam observar diretamente a introdução da chave; ou que mais tarde consigam distinguir as marcas deixadas pelos dedos na superfície de toque. Além disso, estes métodos não são acessíveis a algumas classes de utilizadores, nomeadamente os cegos.

Nesta dissertação propõe-se que os métodos de autenticação em smartphones podem ser melhor adaptados ao contexto móvel. Nomeadamente, que a possibilidade de interagir com o dispositivo de forma inconspícua poderá oferecer aos utilizadores um maior grau de controlo e a capacidade de se auto-protegerem contra a observação do seu código secreto.

Nesse sentido, foi identificada uma modalidade de entrada que não requer o canal visual: sucessões de toques independentes de localização no ecrã táctil. Estes padrões podem assemelhar-se (mas não estão limitados) a ritmos ou código Morse.

A primeira contribuição deste trabalho é uma técnica algorítmica para a deteção destas sucessões de toques, ou *frases de toque*, como chaves de autenticação. Este reconhecedor requer apenas uma demonstração para configuração, o que o distingue de outras abordagens que necessitam de vários exemplos para treinar o algoritmo. O reconhecedor foi avaliado e demonstrou ser preciso e computacionalmente eficiente. Esta contribuição foi enriquecida com o desenvolvimento de uma aplicação Android que demonstra o conceito.

A segunda contribuição é uma exploração de fatores humanos envolvidos no uso de frases de toque para autenticação. É consubstanciada em três estudos com utilizadores, em que o método de autenticação proposto é comparado com as alternativas mais comuns: PIN e o padrão Android.

O primeiro estudo (N=30) compara os três métodos no que que diz respeito à resistência a observação e à

usabilidade, entendida num sentido lato, que inclui a experiência de utilização (UX). Os resultados sugerem que a usabilidade das três abordagens é comparável, e que em condições de observação perfeitas, nos três casos existe grande viabilidade de sucesso para um atacante.

O segundo estudo (N=19) compara novamente os três métodos mas, desta feita, num cenário de autenticação inconspícua. Com efeito, os participantes tentaram introduzir os códigos com o dispositivo situado por baixo de uma mesa, fora do alcance visual. Neste caso, demonstra-se que a autenticação com frases de toque continua a ser usável. Já com as restantes alternativas existe uma diminuição substancial das medidas de usabilidade. Tal sugere que a autenticação por frases de toque suporta a capacidade de interação inconspícua, criando assim a possibilidade de os utilizadores se protegerem contra possíveis atacantes.

O terceiro estudo (N=16) é uma avaliação de usabilidade e aceitação do método de autenticação com utilizadores cegos. Neste estudo, são também elicitadas estratégias de ocultação suportadas pela autenticação por frases de toque. Os resultados sugerem que a técnica é também adequada a estes utilizadores.

Palavras-chave: segurança, usabilidade, móvel, gestos, toque

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Smartphones and tablets have, to a large degree, fulfilled aspirations of ubiquitous computing. They have brought along new ways to work, play and socialize, whenever and wherever. And as they move from our desks to our pockets, computers have become, more than personal, intimate. Mobile devices can be tokens of one's identity - whoever has Alice's smartphone is, in many ways, Alice. That person can send messages to her contacts, access private communication, shop with her credit card, and even know where she has been. General trust in ubiquitous computing cannot be sustained if devices weighting a few dozen grams, that are easily lost or stolen, can enable exposure of private life, impersonation or pervasive surveillance of one's every movement.

In this dissertation, an authentication method for smartphones that aims to mitigate this threat is proposed. This proposal is framed within recent advances in the understanding that human factors play a central role in security, that is, within the field of HCI Security (HCISEC).

1.1 Motivation

Security and privacy risks related to new usage practices are an enduring challenge. As people store more personal data in their mobile devices, the consequences of security failures can become devastating. A typical counter-measure to avoid this risk is to set up a secret code that has to be entered to unlock the device after a period of inactivity. The expectation is that an ill-intentioned party that acquires a person's phone will not be able to use it in any meaningful way, lacking the knowledge to successfully authenticate.

Unlock authentication is, in a sense, the gatekeeper to privacy. But the methods used to authenticate in mobile devices are largely a legacy from the desktop era. Entering passwords is known to consume a non-trivial amount of time and to require significant cognitive effort (Lee & Zhai 2009). Such may be acceptable when sessions are long, but typical mobile sessions are of a different nature – they are typically short,

single application interactions (section 2.2 provides an overview of smartphone usage patterns). It is not reasonable to expect that only but a few zealous users are willing to constantly authenticate in such a fashion (Clarke & Furnell 2005).

Usability, in a broad sense, is an upper bound to security. As Cranor & Garfikel note in the preface of their seminal book "Security and Usability" (2005, p.iv), "a computer that makes you authenticate every five minutes with a password and a fresh drop of blood might be very secure, but nobody would use it". Causing too much of an inconvenience to the user is a sure path to prevent meaningful adoption (Adams & Sasse 1999).

Currently, the leading unlock authentication methods for commodity smartphones rely on PINs, passwords, and, more recently, graphical codes, like Android's pattern unlock, where the user joins points in a 3x3 grid. Although there is some evidence that the latter is a move forward in usability (Zezschwitz et al. 2013), in all cases security is hampered by the fact that these methods resort to a) location-dependent touch input, and b) visual output. This makes them susceptible to even rudimentary attacks, through a) recognition of smudges left on touch surfaces, and b) direct observation of input interaction, respectively.

These methods, by relying on visual feedback, also prohibit adoption by users with permanent or situational sight impairments. In the particular case of blind users, using graphical codes is not possible, and using PINs or passwords requires unreasonable effort. To enter a PIN with a virtual keyboard reader, such as iPhone's VoiceOver[1], one must pass a finger over the screen while a voice reads out the digits underneath. Azenkot et al. (2012) found that even in a group of blind users familiarized with this technology, unlocking took in excess of 7 seconds, which is clearly too cumbersome for casual use. Furthermore, the process of selecting each digit makes it easier for bystanders to discern the PIN. Finally, authenticating in a such a conspicuous way draws attention to the use of assistive technology, potentially leading to feelings of self-consciousness (Shinohara & Wobbrock 2011).

The latter point highlights an aspect of usability that must not be overlooked: the social context. The leading unlock authentication methods, by requiring explicit visual and touch interaction, make it possible that people around a user can observe the code. In this situation, to unlock the device, the user is left with two options: either try to conceal the action, for example covering the device with one hand, and risk being perceived as distrustful of others; or no conceal it, and risk later intrusion.

1 IOS Accessibility, http://www.apple.com/accessibility/ios/

In summary, changes in penetration and usage of mobile devices have highlighted the need for privacy protection. Unlock authentication, being the first step in securing a device, and despite recent advances, remains a challenge. New technology that addresses it should take into consideration usability, security and accessibility concerns, all in the context of the social settings in which ubicomp devices are often used.

1.2 Objectives

The general objective of the research effort here presented is to *provide a mobile, non-visual authentication method that affords inconspicuous behavior*.

Specifically, the focus is on using patterns of finger taps on the device as the input. Rhythmic interaction is a modality that, although little explored, has been identified as useful when the visual channel is not available (Ghomi et al. 2012). The rationale for such a focus is twofold. First, when resorting to this type of input in a smartphone, the screen location in which tapping occurs is irrelevant, rendering smudge attacks immaterial. Secondly, since the user's visual perception is not needed for tapping, the authentication task can more easily be performed inconspicuously.

A fundamental principle underlying this research is “designing with an adoption process in mind” (Grudin 1994; Grudin & Poltrock 2012). This translates into, from the outset, a) creating actual software artifacts that target widespread hardware, namely commodity smartphones, and 2) evaluate working prototypes with users, looking beyond standard usability metrics, into factors that influence acceptance, including the social context.

To that end, the specific objectives are as follows:

1. Develop an authentication technique using tap phrases – patterns of taps on a binary sensor over time, independent of location. This technique should be computationally efficient and require minimal configuration by example, as is the case with PIN's, passwords and Android's graphic code.
2. Deploy unlock authentication software to commodity mobile devices. The architecture of this artifact should mimic current practices, including using the touchscreen as the sensor.
3. Thoroughly evaluate this authentication method in regards to user experience, resilience to shoulder-surfing attacks, and accessibility.

1.3 Contributions

The work here presented encompasses contributions in two axes:

- A data-driven tap pattern recognition technique requiring a single example for configuration.
- An exploration into human factors concerning the use of tapping interaction for authentication in mobile devices.

Regarding the technical contribution, the specific artifacts produced are as follows:

1. An algorithm to match a tap phrase input to a pre-configured template.
2. An Android implementation of this algorithm within a proof-of-concept application.
3. An analysis workbench, which takes traces from user interactions and replays the authentication operations on competing matching algorithms, providing accuracy metrics.

The proof-of-concept Android application is made available in online in Google Play[2] (the source is also available online[3]). Technical details of these contributions are explained in Chapter 3.

The human factors investigation consists of three user studies:

1. A comparative study (N = 30) where tap unlock is compared to PIN and Android's graphic unlock, in regards to user performance, experience, and resilience to shoulder-surfing (or lack thereof). The results suggest that tap unlock is comparable to the leading alternatives, and that all three approaches are highly susceptible to shoulder-surfing. (Chapter 4)
2. A second comparative study (N = 19) where the 3 alternatives are again compared, but this time in a context of inconspicuous authentication, i.e. without visual feedback. Results clearly show that tap authentication is better performing than the alternatives and therefore better suited for sensitive social settings. (Chapter 5)
3. A study with blind users (N = 16) where usability and acceptance of tap unlock was evaluated, and interaction concealment strategies were elicited through role-playing. Results suggest that the technique is adequate for blind users, addressing concerns of usability, security, and conservation of social comfort. (Chapter 6)

2 https://play.google.com/store/apps/developer?id=Diogo+Marques

3 https://github.com/diogomarques/android-tap-phrase-detector

1.4 Publications

In the duration of my masters degree, I co-authored the following publications:

- Diogo Marques, Luís Duarte and Luís Carriço (2012). Privacy and secrecy in ubiquitous text messaging. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion – MobileHCI '12. New York, New York, USA: ACM Press. doi:10.1145/2371664.2371683
- Diogo Marques, Tiago Guerreiro, Luís Duarte and Luís Carriço (2013). "Under the Table: Tap Authentication for Smartphones", Proceedings of British Computing Society Human-Computer Interaction Conference – The Internet of Things XXVII. Uxbridge, UK: British Computer Society.
- João Guerreiro, Daniel Gonçalves, Diogo Marques, Tiago Guerreiro, Hugo Nicolau & Kyle Montague (2013), "The Today and Tomorrow of Braille Learning", poster accepted for publication in ASSETS'13
- H. Nicolau, K. Montague, J. Guerreiro, D. Marques, T. Guerreiro, C. Stewart & V. Hansong (2013) "Augmenting Braille Input through Multitouch Feedback", poster accepted for publication in UIST'13

This first publication presents work that provided initial motivation into supporting inconspicuous behavior and a starting point for the development of our tap phrase recognition software. The second presents initial results of the work presented in this dissertation. The third and fourth are collaborations that emerged in the context of evaluating tap authentication with blind users.

1.5 Work Context

The research leading to this dissertation was conducted in the Large-Scale Informatics Laboratory's (LaSIGE) Human-Computer Interaction and Multimedia Research Team, located in the Department of Informatics, Faculty of Science, of the University of Lisbon.

This dissertation is a result of work supervised by Prof. Luís Carriço, and conducted in collaboration with Luís Duarte and Prof. Tiago Guerreiro, who co-authored previous publications. Therefore, when referring to the work, "we" is used instead of "I".

Chapter 2

Background and Related Work

The present chapter is a critical overview of the state of the art pertaining to this work. First, an account of user behaviors and perception is presented, starting at general accounts of privacy and it's relation to technology, and then honing in on specific practices emerging from the introduction of smartphones. In the final sections, criteria for assessing smartphone authentication methods are proposed and matched against the most widely-adopted technologies and also novel proposals in the literature. The approach taken in presenting the related work is to introduce specific critiques throughout, and use these critiques to articulate the relationships in prior art.

2.1 Privacy and Technology

2.1.1 Perception of Privacy

From the end user perspective, security is a practical problem, that comes into play when one asks the question: “is this system secure enough for what I want to do now?” The question may be very hard to answer, not only because of their limited technical knowledge, but because the way security is implement is often not visible (Wood 1977). Most users are unaware that little information is needed for establishing their identity, being mostly concerned with protecting addresses, driver’s licenses, credit card numbers, and official identity numbers (Zhu et al. 2012). Despite that crossing gender, birth date and zip code is sufficient to uniquely identify 63% of the US population (Golle 2006). Indeed, studies have consistently shown that users misunderstand many security technologies, from browser cues like the security padlock (Dhamija et al. 2006) to guarantees of confidentiality in data circulated in Wi-Fi and mobile data networks (Klasnja et al. 2009; Chin et al. 2012).

Another complication is that privacy means different things to different users. It has been suggested, for instance, that younger adults have a greater desire for fine-grained control over disclosure of personal information than older adults, who are mainly

concerned with official information, such as health or financial records (Kwasny et al. 2008). Gender differences have also been observed: in a recent survey, female participants showed more reluctance to make financial transactions on mobile devices and more acceptance of security technologies (Sieger & Möller 2012).

It has also been observed that privacy sensitivity is guided by individual personality traits. A study of attendance control systems with students in Japan and Australia suggests an effect of the "Big Five" personality traits (openness, conscientiousness, extraversion, agreeableness, neuroticism) on perceptions of security. In particular that newer technologies tend to be looked at with more suspicion, despite no actual differences in security level (Uesugi et al. 2010). Our own survey of text-messaging behavior indicated that worries about observation that one is texting are situational but worries about observation of text message content seem to vary according to the user's personality (Marques et al. 2012).

2.1.2 Privacy as a Social Phenomenon

Privacy is an elusive concept. From an individual perspective, privacy is often understood as the ability to select what others can know about us (Kwasny et al. 2008). Therefore, privacy is inherently dependent on individual inclinations. These inclinations, nevertheless, do not exist in a void: they are influenced by extrinsic factors, such as the social, institutional, or cultural contexts.

A visible consequence of overlooking the social and cultural dimensions is that privacy-related problems still persist in new systems, even those designed with privacy as a key concern. For instance, many operating systems– including the most recent versions of Android OS[4] – now support guest user accounts, a feature designed for protecting privacy in device-sharing situations. Usable and secure as the feature may be, it just may not protect us when a friend asks to "just check my email": logging out of one's account can indicate suspicion, violating an unwritten contract. Privacy is not a static set of preferences, but a social product, the result of a two-way interactive process between self and world (Lehikoinen et al. 2007; Dourish & Anderson 2006). Designing systems that accommodate the transient boundaries of privacy is an active research topic (Barkhuus 2012).

2.1.3 The Cost of Privacy

The fact that users knowingly compromise their privacy, for instance by using the easier possible password that a system accepts (Florencio & Herley 2007), has been investigated as a cost-benefit problem.

4 Android, "What's New", http://www.android.com/whatsnew/

One view is that the neglect of privacy protection in computer systems is "entirely rational from an economic perspective" (Herley 2009). The reasoning is that, in aggregate, following security advice and obeying policies is more costly than the benefit of reducing (not eliminating) the risk of privacy integrity. This is so because the costs in time and effort are very frequent and probable, and the potential benefit is very occasional and improbable. For instance, mandating every user to visually inspecting the URL of every link in every email will create a large aggregate cost. But the aggregate benefit of precluding just one vector for phishing attacks[5] is not that high. Indeed, most security breaches that users experience have relatively low impact. A recent study of smartphone "undesired behavior" (Felt et al. 2012) points out that users mainly have to deal with unsolicited offers (spam, ads) and resource drainage (battery, memory, bandwidth).

At the level of the individual user, demanding attention for security decisions also imposes an opportunity cost, creating negative externalities on every other decision. Böhme & Grossklags (2011) proposes a rationale behind the instant dismissal of security dialogs, in which the user is supposed to make a careful decision, but often doesn't: their attention budget is over-consumed.

Too often a view is taken that users do not protect their privacy because they are lazy. Systems are designed with the premise that if only the users were aware of the risk, they would make decisions that enhance their security; and if the benefits of properly using security features are high enough, they will use them. But since attention and time are scarce, what users *want* and what users *do* are not the same thing (Connelly et al. 2007; Spiekermann et al. 2001). The implication for designers is that imposing excessive costs to users can lead them to make bad decisions, rationally.

2.1.4 Usable Privacy and Security

Insights from social sciences have propelled a new understanding of privacy, one in which the user is the pivot between security and the lack thereof. The study of security as a user-centered design problem is now an established discipline.

Even before personal computing was a reality, the human element was already recognized as central to security. In the classic Saltzer and Schroeder (1975) paper on security principles, the necessity that "the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly" was recognized. The problem of authentication was recognized early on, giving rise to

5 An attack where an ill-intentioned party poses as a trustworthy entity in order to obtain personal information, such as credit card number.

proposals like pass-phrases, as a way to "maximize […] the ease of remembering passwords" (Porter 1982).

Two landmark articles published in 1999 propelled researchers in the field of Human-Computer Interaction (HCI) to investigate the interplay between security and usability from several vantage points. The first, "Why Johnny can't encrypt" (Whitten & Tygar 1999), is a usability evaluation of Pretty Good Privacy, an encryption system that had received great attention. It reveals that, despite its soundness, the system wasn't secure in practice, due to user behavior. In effect, only one third of participants were able to sign and encrypt an email message within 90 minutes. Many though that they had been successful, but weren't in fact. The second landmark study, "Users are not the enemy" (Adams & Sasse 1999) was specifically about password practices. Among the findings, a cardinal insight is that users will find ways to circumvent the most stringent security policies and create usability where there is none – for instance, writing down system-generated optimal passwords.

The field has since matured, receiving contributions from the many disciplines that inform HCI. A growing body of knowledge has been produced and disseminated in the standard HCI outlets, and also specialized forums, most notably the annual ACM Symposium on Usable Privacy and Security (which, at the time of writing of this document, is ranked 9 among HCI publications in Google Scholar[6]). More extensive accounts of the inception and development of the discipline, sometimes dubbed "Usable Privacy & Security" (UPS) or "HCI security" (HCISEC) can be found in Payne & Edwards (2008) and Garfinkel (2005, chap.2). An overarching view of the domains of interest and foundational contributions is available in Cranor & Garfinkel's book (2005).

Efforts in understanding the human element have undermined a somewhat autistic view of IT security, one in which there is a major trade-off between security and usability; the implication being that users need to pay so their privacy can be assured. In fact, security systems must address a broad set of requirements, otherwise they won't be viable. Usability is one of them, and a crucial one at that. If experts were to design an encryption system that was uncrackable but took 13.8 billion years to encode a message using all computational power currently available, one would say that it wasn't viable because it does not address the requirement of efficiency, and not that it should print out the warning "please hold for a universe's lifetime to complete this task". Yet, people are expected to memorize a different password for each online service they use and, adding insult to injury, that each password is lengthy an unintelligible, despite the fact that this is all but humanly possible. As we cannot easily upgrade our brains, usability should be

6 Google Scholar, Top publications - Human Computer Interaction, http://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_humancomputerinteraction

seen as a prerequisite, not an obstacle, for any security system that involves human activity.

2.2 Smartphone Usage

It is hardly a surprising observation that smartphone usage is fundamentally different than that of the desktop computer. These devices come equipped with a plethora of context sensors, allow new interaction techniques, and follow us (or even guide us!) everywhere.

One interesting empirical observation is that interactions with mobile phones are "bursty", that is, very short and clustered. In study of activity logs from 20 users "in the wild", looking at how much time the device screen was on, 83% of interactions were below 40 seconds (Shye et al. 2009). A larger study of 255 Windows Phone users found that in most interactions (90%) only a single application is used (Falaki et al. 2010). This study sheds light on another point: it is still hard to identify the typical user. The average daily number of interactions and interaction time lengths varied by a order of magnitude: 10 to 200 and 10 to 250 seconds. Average data consumption per user/day even more so: from 1 to 1000 MB. However dramatic these variations are, they can be explained by the kind of activities users perform. The study finds that application popularity can be modeled as an exponential distribution. It follows that users who engage in the activities that the most popular applications enable will show overwhelmingly increased usage statistics.

The bursty-ness of usage is revealed not only in interaction frequency and length, but actually in attention spans (Oulasvirta et al. 2005). Mobile interactions are often not a goal but a means; they are intertwined with human activity (e.g. finding a route to a destination) and context (e.g. walking in a rainy day) that require attention. One implication of these observed patterns is that, when designing mobile applications and systems, it is advantageous to optimize for time and cognitive effort, breaking action chains into smaller units, allowing for interruptions.

Recent work about long-running mobile tasks indicates that supporting not only interruption, but *recovery from interruption* is crucial. Brumby & Seyedi (2012) analyzed the impact of device auto-locks on driving performance while writing a message, and found that interruptions caused by the driving context were more taxing to lane-keeping performance when the auto-locks were frequent. Since the cost of recovery was high (unlocking), users opted to focus more on the device and less on the road.

In summary, smartphones present a great opportunities but also a number of limitations when compared with traditional computers. Beyond the obvious battery consumption considerations, or lack of performance of virtual keyboards (Lee & Zhai

2009), users are easily frustrated if systems require them to jump through hoops when they want to complete a simple, often very quick task.

2.3 Smartphone Users' Security Concerns

There is abundant evidence that users worry about security in their smartphones. Even in the pre-iPhone era, a study of factors influencing the choice of a handset indicated that security considerations ranked second, only after battery life (Clarke & Furnell 2005). User concerns over security are preventing them to take full advantage of the technology at their disposal: as many as 70% of users are reluctant to perform many privacy-sensitive tasks on smartphones (Ben-Asher et al. 2011). In comparison to what they do in desktop computers, users worry more about, for instance, disclosing their social security number and health data, using banking services or shopping (Chin et al. 2012).

The concerns of users can be classified, broadly, in three categories: data loss, financial costs, and data exposure.

2.3.1 Data Loss

Concerns with data loss are widely recognized and currently addressed by many popular synchronization and automatic backup systems. Modern smartphone operating systems, in effect, incorporate such functionality – for instance, Apple iCloud[7], and Android's Google Sync[8] and Backup API[9]. Third-party applications for such purposes are also widely disseminated (e.g. Dropbox[10], DataSync[11], Wuala[12]). Nevertheless, recent findings indicate that users are reluctant to backup their data to the "cloud", and also have difficulties in setting-up the appropriate configurations for safeguarding some types of data that they find valuable or sensitive (Muslukhov et al. 2012).

2.3.2 Financial Costs

Smartphones bring along new financial considerations for the user, namely: 1) the cost of the device itself and 2) the cost of using the network infrastructure for voice and data. In a recent survey of concerns about smartphone malware effects, the top 3 user-ranked risks were related to financial costs. First-ranked is the risk of permanently

7 iCloud, http://www.apple.com/icloud/setup/ios.html

8 Google Sync, http://www.google.com/sync/index.html

9 Backup API, http://developer.android.com/guide/topics/data/backup.html

10 Dropbox, http://www.dropbox.com

11 DataSync, https://play.google.com/store/apps/details?id=com.quintstoffers.DataSync

12 Wuala by laCie, http://www.wuala.com/

damaging the device. Second and third were the risks of malware making calls or send messages to services that cost money (Felt et al. 2012). User concerns seem, then, to be raised when financial costs are higher. Indeed, another recent survey indicates that users perceive making international calls as a more security-sensitive task than making local calls, which tend to be cheaper (Ben-Asher et al. 2011).

2.3.3 Data Exposure

Many types of information kept on smartphones can be considered sensitive, in the sense that it would have detrimental effects to the owner if it were to be exposed. In a survey of 465 smartphone users, more than 50% considered stored passwords, files, contacts, emails, text messages, call logs, location traces, schedules, pictures and videos to be sensitive of very sensitive (Ben-Asher et al. 2011). These findings align with the aforementioned study about perceived malware risks (Felt et al. 2012): instances of highly-ranked concerns among users includes malware that shares photos or text messages, changes the PIN/lock pattern, or captures the call log.

Although malware infection can lead to data exposure, violation of privacy is often the product of another person gaining physical access to device (Muslukhov et al. 2013). Unlock authentication provides a defense against this most straightforward threat.

2.4 Smartphone Authentication Methods

For smartphones – the most common private computers –, authentication methods are the de facto gatekeepers to privacy. Typically, in order to save battery, these devices partially shut down after a period of inactivity, and can be set-up so that authentication is required for bringing them back to operation. This has been recognized as great opportunity to protect the user's privacy. Even if the device is lost or stolen, it is highly likely that it will be locked when and if an ill-intentioned party gains physical access, thus limiting privacy risks to the owner.

However, there is a growing realization that authentication methods from the desktop era are unsuited for the mobile context. Passwords and proprietary tokens are such a cause for worry among the technology industry that in 2013 a consortium was launched to tackle the issue: the Fast Identity Alliance[13], which includes Google, PayPal, LG, and others, working under the slogan “Forget Passwords!”.

There has also been a large influx of proposals of novel and exotic authentication methods specifically targeted to ubicomp devices coming from the HCISEC

13 FIDO Alliance, http://fidoalliance.org/

community. Some of these proposals are beginning to echo with the smartphone manufacturers: in May 2013, Motorola executives discussed some of their explorations into touchless authentication, namely using radio-enabled tattoos or pills[14]. Motorola's model Moto X, launched in July 2013, is also able to use any Bluetooth device selected by the user as a token for unlock authentication[15].

In the next subsection, the currently widely-adopted authentication mechanisms will be reviewed. Following, an overview of security risks that they impose are articulated into a general threat model, focused on casual and opportunistic ill intentions. Recent proposals for smartphone authentication methods that attempt to address this type of threat are then organized according to Wood's taxonomy. Finally, recent efforts in making authentication accessible to blind users are presented.

2.4.1 Leading Methods

"No one pretends that democracy is perfect or all-wise. Indeed, it has been said that democracy is the worst form of government except all those other forms that have been tried from time to time", famously said Churchill. Replacing "government" and "democracy" for "passwords" and "authentication" gives an accurate account of the general understanding of authentication methods until recently. Much was tried but, despite the problems, we were more-or-less stuck with passwords.

Passwords, unlike democracy, are so generally despised that finding ways to make them obsolete became the main driver for the emergence of HCISEC. We know that if we assign good passwords to users, they won't be able to memorize them, and will write them down (Adams & Sasse 1999) or use other coping strategies, rendering them insecure. If instead we allows users to choose their own passwords, we can be certain that security will lack, since they will choose sequences that are easy to memorize and quick to enter, and thus susceptible to dictionary attacks[16]. The middle-ground between these two approaches has been imposing password composition policies, e.g. enforcing a minimal number of characters or the mixed use of number and letters. But imposing these policies is a zero-sum game: if they are too restrictive, users typically resort to (insecure) coping strategies, if they are too loose, user will choose easy passwords

14 Motorola's Dennis Woodside and Regina Dugan: The Full D11 Interview, http://allthingsd.com/20130529/motorolas-dennis-woodside-and-regina-dugan-talk-moto-x-tattoos-and-taking-big-risks-at-d11-full-video/

15 Motorola Moto X, http://www.motorola.com/us/shop-all-mobile-phones/Moto-X/FLEXR1.html

16 A type of attack were authenthication attempts are made using large sets of words, including many variations of words in the dictionary and passwords found to be frequent in previous password mass leaks (like the infamous 32 million passwords leaked from the RockYou.com service in 2009).

(Inglesant & Sasse 2010). As a consequence, these policies are often very unrestrictive and therefore largely misguided (Bonneau & Preibusch 2010; Komanduri et al. 2011). The general understanding that passwords offer security is, in fact, a notable case of suspension of disbelief.

Virtually all commercial smartphones now incorporate "secure" unlock features. Vendors offer authentication using Personal Identification Numbers (PIN), which are simply numeric passwords, but more convenient to input (Clarke & Furnell 2005), and also standard alphanumeric passwords. Even in the smartphone era, passwords remain the leading approach to authentication, despite virtual keyboards being even more taxing to passwords authentication performance (Schaub et al. 2012).

There is, however, already significant movement towards other mechanisms. Currently, the only widely-used alternative is Android's pattern unlock. In this method, users are presented with a matrix of 9 points, and must trace a directed path over them with their finger. Recent research (Zezschwitz et al. 2013) indicates that, in the wild, users take more time and make more input errors with this method, and yet still like it better than using PINs. One possible reason for this dichotomy is that PIN entry errors are more adverse, in the sense they cause a non-trivial interruption, raising the cost of recovery (see section 2.2 for a discussion of interruptions and recovery from them).

Recent versions of Android also include a face recognition technique for unlocking. This feature, however, has been widely publicized on the web for being insecure[17], since it is easy to bypass using pictures from the owner, extracted for instance from social media services. This is a known problem of face recognition systems, and stems for the fact that liveness is difficult to detect unobtrusively (Findling & Mayrhofer 2012; Tronci et al. 2011). In our own studies, we failed to found a single user that uses Face Unlock.

Media reports suggest that, at the time of writing of this dissertation, manufacturers Apple and Samsung are preparing to deploy fingerprint readers in their high-end devices, namely in the upcoming iPhone 5S[18] and Galaxy S line[19] models. The manufacturer Motorola launched a model with fingerprint authentication in 2011, the

17 For instance, Wired, "Video: Ice Cream Sandwich Face Unlock Defeated With Photo", http://www.wired.com/gadgetlab/2011/11/video-ice-cream-sandwich-face-unlock-defeated-with-photo/

18 BGR, New iPhone 5S part leak points to fingerprint scanner, http://bgr.com/2013/08/14/iphone-5s-photos-parts-fingerprint-scanner/

19 SamMobile, "HOT: Samsung prepares fingerprint protection", http://www.sammobile.com/2013/05/21/hot-samsung-prepares-fingerprint-protection/

Atrix 4G[20], but has since then discontinued it. It is unclear why the same technology is now expected to solve a problem that it didn't before.

In summary, however inadequate are PINs and passwords, they are currently still the basis for leading smartphone authentication methods. Android's secret “drawing”, however, is already a widely adopted alternative, at least for unlock authentication. These are, at this point in time, the leading methods. Some other attempts have been made by the industry, including using fingerprints and face recognition, but these did not seem to gain much traction.

2.4.2 Casually Insecure: a Threat Model

A major cause for the failure of passwords as a general-purpose form of establishing identity is an ill-defined threat models. Any security feature of a system makes sense only in the context of the types of threats it is securing against. Passwords and respective composition policies have, unfortunately, gained the status of being secure against a mythical general threat model, despite the fact that they are designed to address, in essence, brute force attacks performed by security specialists. Even so, passwords are commonly being cracked through dictionary or brute-force attacks.

But is the threat of cracking important in all systems where passwords are used? The answer is an emphatic no. ATMs, for instance, usually require a card and a 4-digit numeric password; after 3 failures to properly enter the code, further attempts are blocked. Even if an attacker gains access to the card, cracking PIN's by trying random combinations has a vanishing likelihood of success[21]. In effect, ATM breaches are usually the result of surreptitious observation and social engineering performed by con artists, or outright violence by criminals.

When considering the two leading smartphone authentication methods, PIN/password and Android's draw code, even if it was stipulate that brute-force cracking is unfeasible, some attacks are so rudimentary that even the casual user can employ.

Shoulder-surfing Attack

Shoulder-surfing is a direct observation attack where a third-party is able to discern at least some features of a secret code. Although the expression indicates that the

20 Motorola, Atrix 4G, https://motorola-global-portal.custhelp.com/app/product_page/faqs/p/30,6720,7898/

21 It is not impossible that an attacker gains mass access to an ATM system through other means, for instance obtaining the database of user and card data,, and then performing an offline attack but this has little to do with the end-user authentication mechanism.

Figure 1: Oily residuals left on a touchscreen. Left: upon simulated PIN entry. Right: upon simulated Android pattern entry.

third-party is located behind the user, it is commonly used as an umbrella term for situations in which there is the ability to detect details of the interaction directly and surreptitiously. For instance, the observing party might be across the table from the authenticating user.

The ability to infer keyboard input using video from the interaction is well established, both for desktop keyboards (Balzarotti et al. 2008) and for smartphones (Maggi et al. 2011; Schaub et al. 2012), even in realistic, noise-filled settings. For smartphone security, addressing the threat of observation by video-cameras or other human-analog sensors, is, however, less of concern than thwarting shoulder-surfing by humans, given the high mobility of the device and the frequent use in social settings (Church & Oliver 2011), that enable casual observation.

Smudge Attack

When a user interacts with touchscreens it is very likely that oily residues from the fingertips will be transferred to it. This can leave compromising traces that enable an attacker to, at least partially, reconstruct the authentication interaction. This has been called a smudge attack. Aviv et al. (2010), presents the first systematic analysis of the feasibility of such attacks, suggesting that the smudges are very persistent, and usually not wiped off in normal operation, including pocketing the device. Their study focused on Android's pattern unlock, but subsequent work indicates that the same principle applies to virtual keyboards (Zhang et al. 2012), especially numeric keypads for PIN-entry.

Although in these studies images of the devices were captured with photography for a more reliable analysis, smudges from authentication interaction are often visible to the naked eye. This is evident in figure 1, which show unaltered photographs of smudges from pattern and PIN entry. The implication is that, by simple observation, even non-experts can discover (or reduce the space of) a user's password or pattern.

Device Acquisition

Authentication mechanisms often seem to address a threat model wherein highly sophisticated bad guys, after acquiring one's phone and taking it to their lairs, are able to crack them and do us harm. This is very unlikely to happen. But our children, spouses, friends, co-workers, strangers in the subway and coffee-shops, who just happen to have an opportunity to acquire our devices, are very real adversaries. No technical knowledge is actually needed to perform shoulder-surfing or smudge-attacks. Smartphones are used in a social context, and it is in this context the threat must be framed.

2.4.3 New Ways to Establishing Identity

In a seminal study of passwords, Wood (1977) identifies three types of methods by which "a person's identity may be established for the purpose of allowing access to a remote computer system:

- something the person *knows*
- something the person *has*
- something the person *is"*

Although some of the current methods can fall within more than one category[22], this taxonomy largely captures the main trends in smartphone authentication, including recent advances in PIN/password variations, graphical secrets, Bluetooth/NFC tokens, biometrics and many other hybrid proposals. The method proposed in this dissertation falls within the first category, "something a person knows", or knowledge-based authentication (KBA). Others have previously offered systematic overviews of classic and novel authentication methods in the three families, including De Luca (2011, chap.2), Paz (2011, chap.2) and Dunphy (2013, chap.2). Here, the focus is on recent KBA proposals, accounting for:

1. The limitations imposed by smartphone usage, namely the need to optimize for unobtrusiveness to the main objective the user is trying to perform (Adams & Sasse 1999). This means, at the very least, not imposing strong costs in time and effort.
2. The social context in which smartphones are used, namely the threat to security posed by non-expert adversaries in said context, including the degree to which they are protected from smudge and shoulder-surfing attacks, and the ability to allow inconspicuous interaction.

22 For instance, ATMs require "something a person has", a card, and "something a person knows", a PIN.

3. The degree to which these techniques are deployable in existing or foreseeable smartphone platforms. This aligns with an aspect of the motivation for this work, namely the principle of "designing with the adoption process in mind", as enumerated in Chapter 1.

Having the user share a secret with the system is often the most cost-effective way to establish identity, and the basis for the most disseminated methods, like PINs and passwords. Weaknesses in password-reliant methods has motivated the exploration other KBA approaches.

Graphical Passwords

Graphical passwords are a particularly interesting case study, since they have been a great focus of HCISEC research since the inception of the field. An in-depth overview of mechanisms relying on graphical passwords is outside the scope of this work, and can be found, for instance, in Biddle et al. (2012). Usually, graphical passwords are categorized in:

- Drawmetric systems, where users insert a drawing, which is then compared to a template. Examples include Android's pattern unlock and Draw-a-Secret (Jermyn et al. 1999).
- Locimetric systems, where users leverage their recognition of image features, for instance selecting specific locations. Examples include Passpoints (Wiedenbeck et al. 2005) and Windows 8/RT picture password sign-in[23].
- Cognometric systems, where users must only recognize some images, which they previously memorized, from a greater set of images that includes decoys. The most notorious example is Passfaces[24].

Besides Android's pattern unlock, other graphical password-reliant techniques have not found wide uptake in smartphone platforms. Reasons for this, aside from understandable inertia, include:

- Not considering deployability in smartphone platforms as a requirement (Dunphy et al. 2010).
- Many methods being found less secure than claimed after further examination (Biddle et al. 2012). For instance, drawmetric and locimetric mechanisms are often susceptible to smudge attacks, when they require location-specific interaction with the touchscreen.

---

23 Windows, "Sign in with a picture passwords", http://windows.microsoft.com/en-us/windows-8/picture-passwords

24 PassFaces ™, http://www.passfaces.com/

- Unreasonable demand of effort to the user, given smartphone usage patterns.

Regarding the last claim, usability problems in graphical passwords schemes have been previously observed, for instance in Chiasson et al. (2007), but still endure in recent work. One classic measure of usability is task completion time. As per our studies presented in the following chapters, PIN entry consumes approximately 3 seconds. Others have measured it to be approximately 1.5 seconds (Bianchi & Oakley 2012). Differences can be explained by how measurement is performed. Many consider the first key press or touch event to be the beginning of the interaction. In this work, the interaction is considered to begin as soon as the PIN entry screen is shown, thus accounting for the time a user needs to get ready for entry. Following, three examples of recent graphical password techniques that address aspects of the proposed threat model, but impose unfeasible task completion times.

Zakaria et al. (2011) proposed addressing shoulder-surfing protection for drawmetric systems with three obfuscation techniques, finding that only one had simultaneously "reasonable" usability and security. "Reasonable", however, means imposing an average login time for medium-security passwords of 6.5 seconds. WYSWYE, another proposal, but this one addressing shoulder-surfing for cognometric systems, required upwards of 20 seconds (Khot et al. 2012). Recently, von Zezschwitz et al. (von Zezschwitz et al. 2013) proposed Marbles and Marble Gap, two cognometric-like techniques designed to be resilient to smudge attacks. The secret is composed by a sequence of colors, e.g. red-white-blue-yellow. In Marbles, users are presented with a circle of 10 colored marbles an must drag the right sequence to the center; the circle rotates in each interaction so that smudges left have no fixed meaning. In Marble Gap, 20 circles are dispersed through the screen, and must be dragged to a specific area in order; in each interaction the circles are redistributed randomly. Although the techniques are shown to be resilient to smudge-attacks, entry consume on average 6 to 8 seconds. When comparing to 3 seconds, all these techniques are too taxing to users' attention.

Haptic Techniques

Other recent proposals have departed from the graphical password paradigm, and tried to leverage the augmented capabilities provided by smartphone hardware. PhoneLock (Bianchi, Oakley, Kostakos, et al. 2011) is a PIN entry system that uses audio or vibrotactile cues. Since it does not rely solely on visual cues, the system is resilient to shoulder-surfing (in the audio case, observation is prevented with the use of headphones). Furthermore, this system allows eye-free interaction, therefore being well suited for social contexts, and is actually implement in the iPhone platform. The method works by mapping possible PIN digits to audio/tactile cues, having the user lookup the

appropriate cues traversing their finger through a circular interaction area, and finally performing a gesture upon successful recognition of each digit. The cost the user pays is memorizing the cues that map to each digit, which is non-trivial effort. Even if training demands are discounted, authentication was found to take an average of 19.9 seconds in haptic mode and 12.2 seconds in audio mode, likely because of the cognitive effort required to map digits to cues and the need for explicit lookup (Bianchi & Oakley 2012).

SpinLock (Bianchi, Oakley & Kwon 2011) is a related haptic technique, but one where no mapping is required. In this system, as the user moves a finger in the circular interaction area, he/she receives periodical cues, having only to count their number. Each number is also associated with a clockwise or counterclockwise direction, thus expanding the key space. The system gives safeguards against smudge attacks by varying the space between cues, i.e. one time a full spin may give 4 cues and the next 10 cues. However, average interaction times were found to be 13.8 seconds in haptic mode and 10.8 seconds in audio mode, which should again be compared with 3 seconds for traditional PIN entry.

Gaze-based Techniques

One recent trend in KBA systems is using eye-gaze to tackle both shoulder-surfing and smudge attacks. Since the gaze does not require physical contact, smudge attacks are impossible; shoulder-surfing protection is also assured as long as there's no obvious feedback on screen. Observation of eye movement can be a threat in some cases, where a sequence of eye movements is visible to the attacker.

EyePassword (Kumar et al. 2007) was a seminal proposal for password/PIN entry using eye-gaze, in a time where eye trackers were becoming affordable. Although a keypad is proposed, the evaluation only considers alphanumeric keyboards. Login times are between 9 and 12 seconds, depending on keyboard layout and whether key selection is performed by dwelling 450ms on a key or pressing the space bar. De Luca et al. (2007) further explored eye-gaze PIN entry, using a virtual keypad modeled after an ATM interface. They found that login time varied between 12 and 13 seconds, depending on the key selection technique. They also found error rates varying between 20.6% and 23.8%, which is considerable, especially if error recovery is not possible, as is the case (since validation only occurs at the end of entry).

Some gaze-based techniques are extensions of locimetric graphical passwords, i.e. the user is presented with an image (or a sequence of images) and selects some secret locations within it. The selection, instead of requiring clicks or touches, requires only

suspended gaze. Forget et al. (2010) investigated this approach and found it viable. Login time, however, averaged 36.7 seconds in the best performing of two conditions.

One known problem with selecting locations in a picture is that users tend to select obvious hotspots, or salient features of the image. Bulling et al. (2012) propose addressing these issues by automatically detecting these hotspots and preventing the user from selecting them by obfuscating them with a mask. Although the study indicates that the security of the method is increased with this obfuscation, usability remains problematic: only ~25% of users rated it as high, compared with ~75% for eye-gaze PIN entry.

EyePassShapes (De Luca et al. 2009) is an approach similar to drawmetric graphical passwords that uses eye-gaze, specifically eye gestures. A PassShape (De Luca, Weiss & Hussmann 2007) is a graphical secret that maps into a PIN, much like Android's pattern unlock (the 3x3 matrix can be conceived as digits, and the drawing connects them sequentially). For users that selected a PassShape with a single stroke, login took an average of 5.3 seconds, which is promising. The technique was shown to be more resilient to observation than traditional PIN entry, but still suffered from a 55% successful attack rate, when the eye gestures were made visible to others.

As is the case with graphical passwords and haptic techniques, gaze-based techniques imposed extended task completion times, in many cases not suitable for smartphone usage. Although none of the aforementioned methods was implemented in smartphones, in this particular case, they should not be dismissed, given the rapid advances in computer vision/gaze-detection techniques and the diffusion of libraries such as OpenCV[25], that make them viable alternatives in the foreseeable future. Regarding supporting a social context, namely the ability to interact inconspicuously, these methods occupy a middle-ground: in some specific cases, for instance with the phone laying on a table, authentication could be performed without others noticing it.

Implicit Authentication

Some recent authentication techniques have combine elements of the “something you know” to the “something you are” paradigms, without being exactly biometric methods. Instead, they still require a user to enter the secret, but also assess if the way in which input was performed is consistent with previously known parameters measured from the same user. These approaches have advantages in terms of theoretical security, since the key space is expanded, and, conceptually, do not pose much greater usability barriers than the KBA method they are based on. Furthermore, they tend to provide

25 OpenCV, Open Source Computer Vision, http://opencv.org/

resilience to shoulder-surfing and smudge attacks, since the implicit gaging of user behavior can't easily be replicated.

One implicit authentication method is analyzing keystroke dynamics while a user is inserting a password (Bergadano et al. 2002). While this is a somewhat well-known technique for passwords in physical keyboards, one experiment with mobile devices (Hwang et al. 2009) makes the feasibility of smartphone deployability questionable. To use keystroke dynamic effectively for PIN entry, users had to be forced into an artificial rhythm, which in turn raises the effort required of them.

More promising results were, however, obtained from gaging implicit behavior while using Android's pattern unlock. De Luca et al. (2012) finds that by extracting features like pressure, time and speed of touch events, the graphical password can be inserted with no increased effort to the user but with added security, including protection from observation and smudge attacks. The approach, however, is parametric, that is, the users had to create a model by exemplifying multiple times at enrollment. Furthermore, the system also denied access when a legitimate user was trying to authenticate ~19% of the time, which is rather high, i.e. in one of every five attempts users could not login. It should be noted that some users performed much better than the average, suggesting that there is room for substantive improvement. A similar observation was made in our studies, suggesting that learning the interaction technique may improve results. In this case, however, the interaction technique itself is not new, so expecting improvements from learning effects contradicts, in a sense, the concept of implicit authentication.

Rhythmic Authentication

The work presented in this dissertation consists of a method that uses tapping patterns on a touchscreen for authentication. This approach builds upon previous work on rhythmic interaction in general, and rhythmic authentication in particular, although we don't impose any measure or motif constraints on the pattern. In other words, unintelligible tap phrases or ones resembling Morse code are encompassed.

Ghomi et al. (2012) attempts to generalize an input method based on rhythm, making use of both taps and breaks. In this work the types of words that can exist are bounded to 5 – three varieties of tap and two of break, varying in canonical duration. Using fixed durations and threshold-based approaches, in our own experience (Marques et al. 2012), leads to very weak matching, and is thus unsuitable for authentication or other critical interactions.

RhythmLink (Lin et al. 2011), although targeted to peripheral pairing, also relies on detection of rhythmic input. This modality is argued to be well suited for interacting

with devices that have limited I/O capabilities, since only a binary sensor is needed. Several examples are required to train the recognizer which, if applied to an authentication system, would bloat the required effort for configuration. The algorithm does not fully take into consideration tap and break time spans, instead measuring only the distance between each tap. This makes it suitable for devices in which tapping a binary sensor fires events on press but not on release, although it implies losing some of the richness of tap phrases.

The main proposal in which the present work builds upon is TapSongs (Wobbrock 2009), a system were users can configure tapping patterns representing songs. Although the paradigm is similar, TapSongs requires several repetitions on configuration. We also expand on this work by exploring the usability of tap-based authentication in comparison to the alternatives, within the context of a well-defined threat model.

Authentication using rhythms, or tap patterns, addresses the three dimensions identified in section 2.4.3, respectively:

1. Since the theoretical key space is limited only by the resolution of the detection technique, tap phrases can be created that are very short and yet very distinct. The implication is that the user can configure a very short pattern, and therefore limit the usability barrier for him/herself.
2. Tap phrase authentication can be performed without resorting to the visual channel, and, as we'll show, is well suited for social contexts in the sense that it affords inconspicuous interaction. This allows the user a way to prevent observation from casual adversaries when he/she feels threatened. Furthermore, since the location on the screen where the tapping occurs is irrelevant, smudge attacks are, in practice, impossible.
3. As we'll show, tap phrase authentication can be deployed to current smartphone platforms very efficiently.

2.4.4 Towards Accessible Knowledge-based Authentication

One advantage of resorting to eyes-free interaction methods is that they very often can address problems of users with visual impairments. Working on accessible technologies is a staple of the research team within which the work here presented was conducted. In consequence, the opportunity to evaluate tapping authentication with blind users was rapidly identified.

Although limited, authentication for blind users has been previously addressed. ATMs, for instance, often offer auditory assistance, provided the user connects

headphones. Mobile phones with T9 keyboards are also easy to navigate using only touch.

But smartphones typically have touchscreens with virtual keyboards. At least one authentication method has been proposed for these devices: PassChords (Azenkot et al. 2012). With this technique, users input a secret that is a sequence of multi-finger touches, using the 4-finger chord analogy which is already familiar to individuals who write Braille, possibly enhancing recall in those cases. Entering PassChord consumes very little time, in average 2.67 seconds, when comparing to PIN entry with a screen reader (7.52s), and even PIN entry and other KBA techniques for non-blind users. The technique also offers reasonable protection from smudge attacks, given that in the calibration phase all 4 fingers are pressed down. The fact that entry requires 4 fingers has two adverse implication: first, it makes it more difficult to authenticate with a single hand, limiting the capacity for concealment of interaction (e.g. perform authentication inside the pocket); and secondly it requires a smartphone that can detect 4 fingers simultaneously, which is not the case with cheaper commodity devices.

Shinohara (2010) summarizes interviews to 19 users of assistive technologies in social and professional settings, finding that they were very aware that using special devices marked them in the eyes of others, preventing them from blending with landscape, and thus causing negative feelings. As a response, Shinohara & Wobbrock (2011) proposes that assistive technology should be designed for social acceptability or, even better, that mainstream technology should have accessibility built-in, i.e. be inclusive, as to avoid necessary unease and feelings of self-consciousness. This is one aspect in which our approach has an advantage over the (yet few) alternatives that have been considered for blind users: it does not require adaptation.

2.5 Summary

This chapter frames our research in a larger context. It starts by contextualizing findings in the social sciences that provide pieces of the puzzle that is privacy. There is now an understanding that privacy is not an immovable concept, but one rooted in individual preferences, which in turn respond to a larger context. One interesting vantage point to understand why people make poor choices when it comes to preserving their own privacy is a cost-benefit analysis. Many of the seminal findings in HCISEC can be explained within this framework. Users choose poor passwords because memory is costly. Users don't encrypt their emails because understanding how to do it correctly is costly.

The introduction of smartphones has heightened the problem of preserving privacy. As Bell & Dourish (2006) eloquently put it, “The ubicomp world was meant to be clean

and orderly; it turns out instead to be a messy one.” Using smartphones creates situations and practices where our intellectual resources are even more constrained, and where new risks emerge. The threat of forced exposure has become more pernicious as smartphones became intertwined with our intimate selves.

Some recourse can be found in the ability our devices have to lock out people that can't establish themselves as the owner. But are the ways we authenticate ourselves appropriate? The current widely adopted authentication methods are susceptible to rudimentary attacks by untrained users, namely shoulder-surfing and smudge attacks. The fact that smartphones are made to be used anywhere, not uncommonly in social settings, is of special concern.

In this context, recent proposals for smartphone knowledge-based authentication methods are overviewed. We find that, when this threat model is addressed, generally it is at the cost of the user's convenience. This trade-off isn't mandatory. Rhythmic authentication is a proposition that has the potential do address both constraints. Furthermore, rhythmic interaction is a modality that can be employed by blind users, potentially enlarging the accessibility frontier of this approach.

Chapter 3

A Tap Phrase Recognizer for Authentication

This chapter introduces a simple, efficient and accurate tap phrase recognizer based on template-matching. In essence, it allows for a user (or developer) to create a tap phrase by exemplifying it a single time; and, at runtime, recognize if new tap phrases match the template. The recognizer is targeted at the mobile authentication scenario, which informs both the choices made in the algorithm design (section 3.2) and the accuracy evaluation procedure (section 3.3). To demonstrate the feasibility of the technique, the recognizer was implemented as proof-of-concept tap phrase authentication Android app (section 3.4).

3.1 Background

Modern smartphones commonly resort to touch input and on-screen gestures as principal interaction styles. One class of gestures that is widely used is tapping. While keystrokes are usually perceived as single events, taps have an implied duration in time. Single taps, long presses and double taps are some examples. These simplest of patterns can be detected efficiently with crude algorithms, that rely solely on timers. For example:

- A tap is long if the release event does not happen within a certain amount of time from the touch event;
- A single tap can be distinguished from a double tap by starting a timer after the release and observing if a second touch event doesn't occur before it finishes.

Tap phrases can be more generally defined as sequences whose words are intervals of “on” and “off” timespans. This definition can be seen as also subsuming Morse code and rhythmic patterns. In Morse, “on” members have one of two fixed sizes (dot or dash), as do “off” ones (spaces between letters or between words). In rhythmic patterns the “on” and “off” intervals are arranged to fit a musical motif.

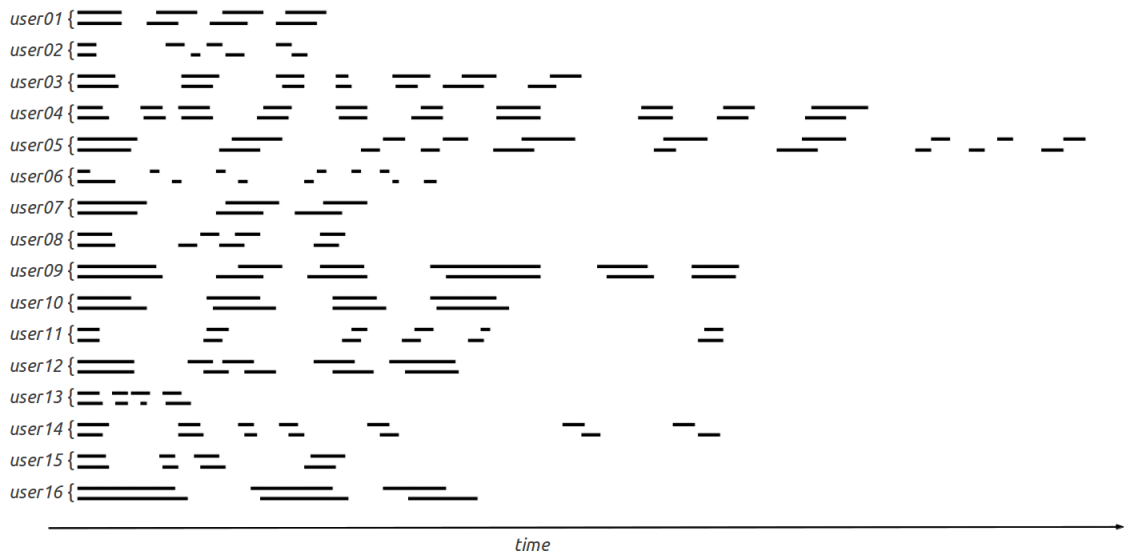

Figure 2: Sixteen cases of tap phrases chosen by end users. Each pair represents a users' template and subsequent repetition, highlighting the actual differences in tap phrases that users perceive as equal. "On" words are represented as black lines and "off" words as the intervals between them.

Operationally, tap phrases are composed by the ordered time distances within and between taps. In a touchscreen, starting with an area that is at rest, a tap phrase begins with the first touch event. When the screen is released, the first tap is finished, having lasted an amount of time t_1^{on}, which is the first word. At the same instant, the first interval between taps begins. At the next touch event, this interval (word t_2^{off}) is finished, and so on, until the screen is left again at rest. This would indicate that a tap phrase is a sequence of words $\{t^1_{on}, t^2_{off}, ...\}$. However, the screen being left at rest, and the interaction ending, is uncertain at the time of the last release event. It is impossible to tell if a new "off" word has started or if the interaction is finished. For practical purposes, an additional constraint is therefore placed on this sequence: it has to start and finish with a tap, being of the form $\{t_1^{on}, t_2^{off}, ..., t_n^{on}\}$.

For humans, comparing two pieces of Morse code, two rhythms, or two tap phrases, is a simple enough task, provided some training. But two tap phrases that a human identifies as being equal may have a great deal of variation. Figure 2 shows a representation of tap phrases through time. Each pair represents two phrases entered by the same user, and both are perceived by this user as being equal. It is clear, however, that they are not.

To match tap phrases, an algorithm is needed that is precise enough as to reject inputs that are too different from the original template, while at the same time not being

so restrictive that the user experiences rejections even when perceiving the phrase to be equal to the template.

3.2 Tap Phrase Matching

Our proposal to matching follows a commonly used approach: first, synthesize a representation of tap phrases that maintains the richness of the raw input while allowing efficient computation; and then perform comparisons using adaptations of well-known similarity metrics and further optimizations. In the following subsections, we define a class of recognizers that follow this prescription and identify what are the variables that can be manipulated to produce specific instances that perform well in the authentication scenario.

3.2.1 Non-functional Requirements

Usability and security are obvious requirements of software tools developed in the scope of this work. But developing a recognition technique that is suited for authentication and performs well on smartphones brings additional constraints that we wanted to address from the outset.

First, smartphones are limited in battery and computational power, so special attention to performance must be taken, otherwise sound theoretical approaches may be unfeasible in real devices, violating our principle of “designing with the adoption process in mind”.

Secondly, as Li (2010) points out, “it is hard to foresee what gestures an end user would specify and what the distribution of these gestures will look like”, indicating that parametric approaches, in which classification of inputs relies on statistical properties of several examples, are not suited. Beyond this abstract consideration, using several examples for configuration creates usability and security problems in the authentication scenario. If the user as to enter the same tap phrase several times for configuration, the process will be very time-consuming. This, in turn, creates an incentive for users to choose weak tap phrases and, furthermore, to not change them frequently.

Therefore, two major non-functional requirements to our classifier are imposed:

1) It must be purely data-driven, with a single example acting as a template, and;

2) It must be able to run smoothly on commodity smartphones.

These requirements are the source of many of the design choices that are explained in more details in the next subsections.

3.2.2 Representation of Inputs

The first step of the proposed technique is generating comparable representations of tap phrases that are machine-friendly and still hold enough information for distinctions to be made accurately. These representations are composed of features described in the next subsections.

Feature #1: Bit Array

The recognizer first digitizes the raw tap phrases into bit arrays. In them, bits are set to 1 for each time unit in which the user was pressing the touch screen, and left at 0 otherwise.

Two reasons informed the choice of this representation. First, it allows the use of common logic operation over two inputs, which is a requirement of many similarity metrics. Secondly, Java, the language in which the Android demo was developed, already offers a compact representation of bit arrays in its standard library, in the form of the BitSet class[26]. This class also already has methods for efficiently performing the aforementioned logic operations[27].

The underlying Android OS provides resolution for touch events at the order of the millisecond. This same resolution can be approximated by the bit arrays. But since it is very unlikely that both the template and the candidate input have the exact same total time, one of them will need to be compressed in order to get bit arrays of the same length (and therefore comparable).

This is achieved by, at runtime, setting the bit array size to the minimum between the candidate's and input's total time (the sum of the words in the tap phrase). For instance, if a template lasts 1000ms and the candidate input 1020ms, both will be represented by bit arrays of size 1000. Having calculated the size, the bit arrays are populated through a sampling process. First, the total time of each tap phrase is divided by the bit array size, obtaining a period (for one of them it will be surely 1). Then,

---

26 BitSet javadoc, Android Developers Reference, http://developer.android.com/reference/java/util/BitSet.html

27 Aside from logic operators, some metrics require the size of the bit array. In Java's BitSet implementation, calling size() will return the length of a vector used internally to keep state, and not the size requested at creation. This can be easily solved by extending the class so it behaves has expected. See https://raw.github.com/diogomarques/onoff-similarity/master/src/net/diogomarques/similarity/FixedBitSet.java

progressing through the tap phrase, at each period set the corresponding element in the bit array to 1 or 0, depending if it falls within an "on" or "off" interval[28].

Feature #2: Total Time

Although the bit array representation, given enough resolution, contains information about the "on" and "off" words, certain important features are necessarily lost. The process of compressing either the template or the candidate input by itself eliminates the information about the total entry time of one of them. Notice that, for any given tap phrase, any other phrase where the words are symmetrically proportional will have the same bit array representation. For example, $\{200^{on}, 200^{off}, 200^{on}, 200^{off}, 200^{on}\}$ has the same bit array representation as $\{400^{on}, 400^{off}, 400^{on}, 400^{off}, 400^{on}\}$, although they are quite different. To address this issue, the total time of the tap phrase is added as a feature of the representation.

Feature #3: Number of Taps

When there is compression, a characteristic as important as the number of taps may be underrepresented in the bit array. If the tap phrase that is compressed has comparatively short taps or pauses, compression may indeed completely remove any information about them, given the sampling process. To address this issue, a third and final feature is explicitly added to the representation: the number of taps.

In summary, the complete representation of a tap phrase is a triplet in the form *(bit array, total time, number of taps)*.

3.2.3 Matching

Our approach to finding a well-performing algorithm is rooted in defining a general class of recognizers that share the same logic, and then evaluating the accuracy of instances produced by different sets of parameters. In this sub-section, we identify the underlying logic in all instances and identify the values that can be manipulated.

Bit Array Similarity Measurement

Once tap phrases are represented as bit arrays of equal length, they can be compared using a number of similarity metrics. We explored four standard metrics, instrumenting them so that they have fixed semantics, namely:

- 0 means completely different, and

28 The Java implementation of this logic can be found in https://raw.github.com/diogomarques/onoff-similarity/master/src/net/diogomarques/similarity/TimePatternDigitizer.java

- 1 means “completely” equal.

The tentative metrics considered were the cosine similarity[29], the Dice-Sorensen coefficient[30], the Jaccard / Tanimoto index[31], and the complement of the Hamming similarity[32]. These metrics are widely used for comparing vectors of binary features in fields like information theory, cryptography and biology. For instance, the Hamming distance was initially developed for error correction in telecommunication, and the Jaccard index was devised as a statistic for plant comparison.

The suitability of these metrics to the tap phrase authentication problem was left as an empirical question, to be addressed in evaluating instances of recognizers that use them.

The definitions of similarity metrics for binary vectors are sometimes vague and subject to different interpretations. Following, precise definitions of the operators and tentative metrics, as they are understood in the scope of this work, are presented.

Operators over the bit array representations are defined as follows:

- The *size* of a bit array is the count of 0’s and 1’s in it contained.
- The *cardinality* of a bit array is the count of 1’s in it contained.
- Binary logic operators AND, OR and XOR can be applied to bit arrays, resulting in a new bit array with equal size, whose members are the result of applying the operation bitwise.

Given two bit sequences A and B of equal length, which are representations of tap phrases in which “1” stands for an “on” time unit and “0” stands for an “off” time unit:

- The function *Cosine*, derived from the cosine similarity, measures how the amount of coinciding “on” time units in A and B relates to the geometric mean of “on” time units in the representations, and is defined as

$$\mathrm{Cosine}(A, B) = \frac{\mathrm{cardinality}(A \,\mathrm{AND}\, B)}{\sqrt{\mathrm{cardinality}(A) \times \mathrm{cardinality}(B)}}$$

- The function *Dice*, which is the Dice-Sorensen coefficient, measures how the amount of coinciding “on” time units in A and B relates to the arithmetic mean of “on” time units in the representations, and is defined as

29 Wikipedia, Cosine similarity, http://en.wikipedia.org/wiki/Cosine_similarity

30 Wikipedia, Sorensen-Dice coefficient, http://en.wikipedia.org/wiki/Dice%27s_coefficient

31 Wikipedia, Jaccard index, http://en.wikipedia.org/wiki/Jaccard_index

32 Wikipedia, Hamming distance, http://en.wikipedia.org/wiki/Hamming_distance

$$\text{Dice}(A, B) = \frac{2 \times \text{cardinality}(A \text{ AND } B)}{\text{cardinality}(A) + \text{cardinality}(B)}$$

- The function *ComplHamming*, adapted from the Hamming distance, measures the complement of the ratio between the quantity of time units where “on” and “off” do not coincide and the total number of time units in the representations, and is defined as

$$\text{ComplHamming}(A, B) = 1 - \frac{\text{cardinality}(A \text{ XOR } B)}{\text{size}(A)} = 1 - \frac{\text{cardinality}(A \text{ XOR } B)}{\text{size}(B)}$$

- The function *Jaccard*, derived from the Jaccard / Tanimoto index, measures how the quantity of coinciding “on” time units in A and B relates to the total number of “on” time units occurring in either A or B, and is defined as

$$\text{Jaccard}(A, B) = \frac{\text{cardinality}(A \text{ AND } B)}{\text{cardinality}(A \text{ OR } B)}$$

The metrics are themselves a parameter for the recognizer instances.

The Decision Threshold

The decision threshold is the minimum level of similarity above which template and input are considered to match. Since similarity is between 0 and 1, this parameter must also be in that range.

Controlling Input Time and Number of Taps

Aside from the bit array similarity, the final similarity function accounts for the number of taps and total time length variations. This is done by placing two controls before the bit array similarity metric is applied, rejecting the user when:

- the number of taps does not match or
- the template and input tap phrases have very different total time.

To verify if template and input are very different time-wise, another threshold is defined. This threshold is a decimal value above or equal to 0.0, and represents the proportion of allowed time variation in relation the minimum between the total times of template and input. For instance, if the threshold is .2, the template's total time is

1000ms, and the candidate's total times if 1020ms, a subsequent input will be rejected if it's lower than 800ms or greater than 1200ms. This threshold of allowed time variation is also a parameter in recognizer instances.

3.2.4 General Recognizer Algorithm

This class of recognizers can be modeled like an algorithm, in which the execution result is dependent on the values of the parameters ascribed to each of the mechanism defined in the previous subsection. Each instance of the recognizer is, in effect, a different matching algorithm, identified by the set of parameters.

The recognizer can be summarized as a sequence of 4 steps:

1. Transform raw tap phrases representing a template and an input into the representations as a triple.
2. Reject the user in the cases where the number of taps does not match.
3. Reject the user in cases where the difference in total time lengths of template and candidate input are beyond the allowed time variation threshold (parameter *atv*).
4. Otherwise, calculate the bit array similarity as measured by a given similarity function (parameter **sm*) and:
    a. Reject the user if it is below the decision threshold (parameter *dt*).
    b. Otherwise, accept.

In pseudo-code:

```
function match(raw_template, raw_input, atv, *sm, dt)
   template := transform(raw_template)
   input := transform(raw_input)
   if template.n_taps != input.n_taps
          reject()
   else if abs(template.total_time – input.total_time)
                              > atv * min(template.total_time, input.total_time)
                reject()
   else
          if sm(template.bit_vector, input.bit_vector) < dt
                reject()
          else
                accept()
```

3.3 Accuracy Evaluation

This section presents results of an empirical evaluation that was performed with the objective of determining parameter combinations that yield well-performing recognizer instances. The performance of a classifier instance is understood in terms of its capacity to correctly match a candidate input to a template. What is being evaluated is not the interaction style or the authentication method, but solely the degree to which algorithms work as expected.

In a matching algorithm, performance is usually defined in terms of accuracy, which is measured by the number of errors it produces (Griaule Biometrics 2009). Errors in matching are of two kinds:

1. The algorithm erroneously rejects an input. In authentication, this usually means rejecting a genuine authentication attempt by the user. These errors tend to increase when the matching technique is too strict. We will refer to these errors as false rejections (FR), but they are also commonly referred to as false non-match errors (FNME).
2. The algorithm erroneously accepts an input. In authentication, this usually means accepting an impostor's authentication attempt. These errors tend to increase when the matching technique is too lenient. We will refer to these errors as false acceptances (FA), but they are also commonly referred to as false match errors (FME).

A well-performing algorithm is, then, one in which both these errors occur infrequently. This is a non-trivial objective to attain since errors vary in opposite directions according to the strictness of matching.

3.3.1 Method

Datasets

The evaluation here presented uses a dataset extracted from the user study described in chapter 4, in which users first configured tap phrases and then authenticated by repeating them. The Android app kept logs of these interactions in XML files. For this evaluation, the data extracted was: 1) the template each user recorded, represented as a sequence of time intervals; and 2) each user's last authentication attempt, represented in the same way. There were only three instances where the user failed to authenticate in the first attempt, but they were noticeable cases of entry errors, since a) the number of taps did not coincide, and b) they were able to authenticate in a subsequent attempt. The criteria of extracting only the last attempt excludes these cases, which are not pertinent for accuracy.

In the initial phases of development, a small dataset, containing the logs of repeated interactions of 3 users was used to develop a matching algorithm that would work "well-enough". This algorithm was implemented in the Android application used in the subsequent user studies[33]. This dataset, however, is not adequate, given that it is very small and that the data collection procedure required the users to perform several repetitions at speeds different than that of pre-configured templates. Since that at the time of writing of this document better datasets were collected, the evaluation was performed on the larger of these.

Apparatus & Procedure

The selected approach to assessing accuracy is exploratory data analysis (EDA). Since the algorithm has many moving parts and parameters that interact with one another, we found model creation and hypothesis testing to be less productive approach than EDA.

To retrieve metrics, we implemented a simulation tool in Java that is able to replay user interactions given the logs. It can also instrument the logic of the algorithm to bypass steps the execution, namely to isolate the effects of enforcing equal number of taps and/or time variation thresholds (which are independent from the metric and decision threshold) from the similarity calculation. The tool automates the process of running the dataset through one or many instances of the recognizer.

The parameters that were used are shown in Table 1. When the class of algorithm is instrumented to not enforce one of its three steps, some parameters are irrelevant. For instance, the allowed time variation parameter set is irrelevant when the algorithm is instrumented to not enforce the time variation control. This will be made clear when discussing results.

Parameter	Levels	# Levels
Similarity metric	Cosine, Dice, ComplHamming, Jaccard	4
Allowed time variation (ATV)	0.5, 0.10, 0.15, …, 0.45	9
Decision threshold (DT)	0.01, 0.02, 0.03, …, 1.00	99

Table 1: Summary of parameters and respective level sets used in accuracy assessments.

33 This early implementation is explained in Marques et al. (2013)

Metrics

For each recognizer instance, errors were measured in the following way:

- Each tap phrase input used in a unlock attempt is matched against the template from the same user. If the recognizer rejects the authentication attempt, this is considered a false rejection. Since there are 30 pairs of template/candidates, for each instance of the classifier, 30 simulations of "genuine" authentication attempts are performed.
- Each template is matched against the templates of every other user. If the recognizer accepts, this is considered a false acceptance. Since there are 30 templates, for each instance of the classifier, C^{30}_2=435 simulations of "impostor" authentication attempts are performed.

The outputs of these simulations that are used for EDA are, for each recognizer instance:

- The similarity score of every simulation in which the input was accepted.
- FRR - the false rejection rate (number of rejections of genuine authentication attempts over 30).
- FAR - the false acceptance rate (number of acceptances of impostor authentication attempts over 435).
- DOT - the dot product of the pair (FRR, FAR) with (0, 0), which summarizes the objective of minimizing both types of errors.

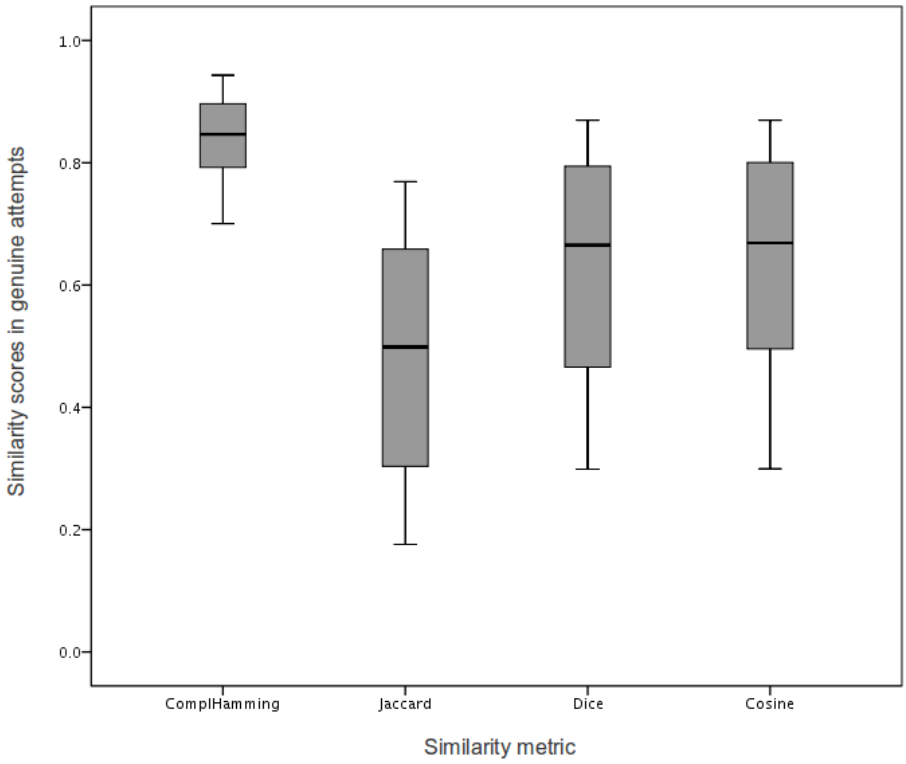

Figure 3: Distribution of scores in genuine authentication attempts. Recognizer instrumented not to control for differences in number of taps and total time.

3.3.2 Results

Similarity Metric Function

Analyzing the similarity metric functions is the first step to understand if the algorithm is viable. To do this, we first look at the distribution of similarity scores for genuine authentication attempts.

To isolate the effect of the metric, four variants of the algorithm were used, each one instantiated with one metric. All instances were instrumented to not control for number of taps or entry time variations.

Figure 3 shows the distribution of similarity scores for each metric in genuine authentication attempts. The first observation that can be made is that the ComplHamming seems to better map to the semantic values of 1 and 0. The average ComplHamming score was .836 (SD=0.069), which approximates "completely equal" better than Jaccard (M=.483, SD=.185), Dice (M=.630, SD=.176) and Cosine (M=.633, SD=.175). The ComplHamming scores are also much less dispersed, with IQR=.109, which compares with Jaccard IQR=.363, Dice IQR=.337 and Cosine IQR=.319.

Function	Arguments	Similarity
ComplHamming	(dense template, candidate w/ 1 error)	**.94**
	(sparse template, candidate w/ 1 error)	**.94**
	(dense template, candidate w/5 errors)	**.68**
	(sparse template, candidate w/ 5 errors)	**.68**
Cosine	(dense template, candidate w/ 1 error)	.95
	(sparse template, candidate w/ 1 error)	.82
	(dense template, candidate w/ 5 errors)	.74
	(sparse template, candidate w/ 5 errors)	.29
Dice	(dense template, candidate w/ 1 error)	.95
	(sparse template, candidate w/ 1 error)	.80
	(dense template, candidate w/ 5 errors)	.74
	(sparse template, candidate w/ 5 errors)	.28
Jaccard	(dense template, candidate w/ 1 error)	.90
	(sparse template, candidate w/ 1 error)	.67
	(dense template, candidate 5 w/ errors)	.58
	(sparse template, candidate 5 w/ errors)	.17

Dense template: 1100011100001111
Dense candidate input with 1 error: 1000011100001111
Dense candidate input with 5 errors: 1000000110011111
Sparse template: 1000001000000001
Sparse candidate input with 1 error: 0000001000000001
Sparse candidate input with 5 errors: 0100000100000011

Table 2: Examples of the effect of the bit array density on the similarity function's results. Functions other than ComplHamming are more sensitive to mismatches in sparse arrays.

Low dispersion is desirable because it makes it easier to identify a small range of decision thresholds that allows most genuine authentication attempts to be accepted. This is particularly of concern because the similarity threshold is, in practice, the greatest obstacle to genuine authentication, since a user knows the number of taps and the approximate length of his tap phrase. For an impostor, controlling for these two factors already blocks many attempts, and similarity score is the last resort. Therefore, if there is much dispersion, the genuine user will likely be burdened with many false rejections, which is highly undesirable since it hinders usability.

Given the visible differences in dispersion of similarity scores yielded by the similarity functions, we hypothesized that there is a fundamental difference in their sensitivity to the ratio of zeros and ones in the bit array. We found that there is, in fact, evidence for such phenomenon. Table 2 illustrates the effect on each metric of having mismatches in both sparse and compact arrays through two examples, showing that ComplHamming, unlike the other measures, is equally sensitive to differences between input and template whether the array is sparse or not.

Since it is difficult to tell what kind of tap phrases users will select “in the wild”, namely if they will be have long intervals and short taps, long taps and short intervals, or anything in between, we conclude that the other metrics are not adequate. The

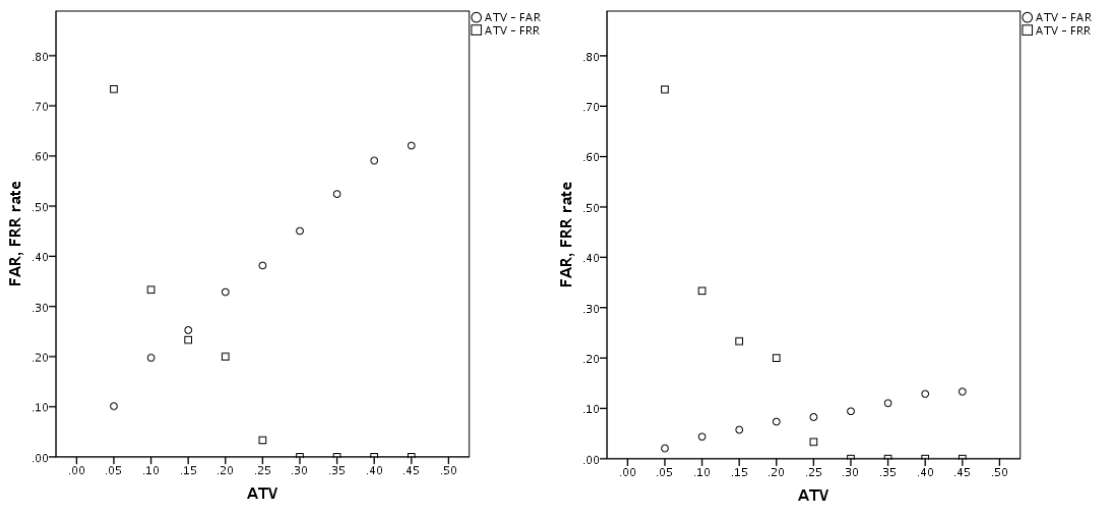

Figure 4: False acceptance and rejection rates as a function of the allowed time variation between template and candidate input, controlling (right) or not controlling (left) for equal number of taps. Similarity metric and decision threshold not applied.

ComplHamming function, unlike the others, is able to distinguishing among tap phrases regardless of the distribution and length of "on" and "off" words.

Controlling Time and Number of Taps

The first measure in place to deny access to impostors is to control for the number of taps and the total time difference between template and candidate input. To assess the effect of these controls, which are the first two actionable steps in the class of recognizers, we first assess the effect of each control separately, and then in conjunction.

To assess the effect of controlling only the number of taps, a single instance of the recognizer was used, in which only the first step was executed, with the result being 0 if the number of taps did not match, and 1 if they did. Running all inputs through this variant gave a false acceptance rate (FAR) of 18.16%, and no false rejections. This means that controlling the number of taps for itself was already filtering out 81.84% of impostor attempts.

The effect of controlling only for total entry time was assessed by creating 9 instances of the recognizer, each parametrized with a different ATV value. In this case, the algorithm was instrumented to not control for the number of taps, and to return 1 in case input and template entry times were within bounds, and 0 otherwise. Figure 4 (left), shows the result on the FRR and FAR of the various levels of ATV. At the .15 level, the FA and FR rates were the closest, with 23.33% of genuine authentication

attempts refused, and 25.29% of impostor attempts accepted. For ATV equal or greater than .30, there were no false rejections, while the false acceptance rate was kept at 45.05 %.

Both the controls for total time variation and number of taps created obstacles for impostor attempts, but the former also had an impact on genuine attempts when set too high. Since the recognizer uses both controls, we next assessed the effect on accuracy of the two controls in conjunction. To that effect, also nine instances of the recognizer were used, this time instrument to execute the first two steps, but outputting 0 if any of the controls failed, and 1 if none failed. Figure 4 (right) shows that controlling for taps and time variations shifted the false acceptance rates downwards while maintaining the false rejections constant. As a result, only using these two controls, it was possible, at the .25 ATV level, to have the FAR at 8.27% and the FRR at 3.33%; or, at the .30 level, completely eliminate false rejections while keeping the FAR at 9.43%.

The implication is that, even before controlling for similarity, and in the best case scenario where one user configured a tap phrase with the same number of taps than others, this tap phrase would only be accepted in approximately 1 out of each 10 of others' devices.

Putting It All Together

Having identified the ComplHamming similarity function as the most adequate, and established that controlling for number of taps and entry time variation already did filter out many impostor attempts, without necessarily imposing to the user a great number of false rejections, we then combined both aspects. To that end, instances of the recognizer that progress through all the steps were created. These instances used the 99 decision thresholds and 9 allowed time variation levels defined in table 1, i.e. 891 recognizers were simulated.

Figure 5 shows, for each level of ATV, the FAR and FRR as functions of the decision threshold. As the results for the allowed time variation effects suggested, at the .25 ATV levels the lines converged, indicating this was the minimum value for which both rates could be kept low simultaneously. Since it is difficult to discern in the graphs what were the best performing recognizers, the 10 instances where the DOT metric was lower are shown in Table 3. It is clear the the top-performing instances had ATV set to between 0.25 and 0.35, and DT between 0.67 and .7.

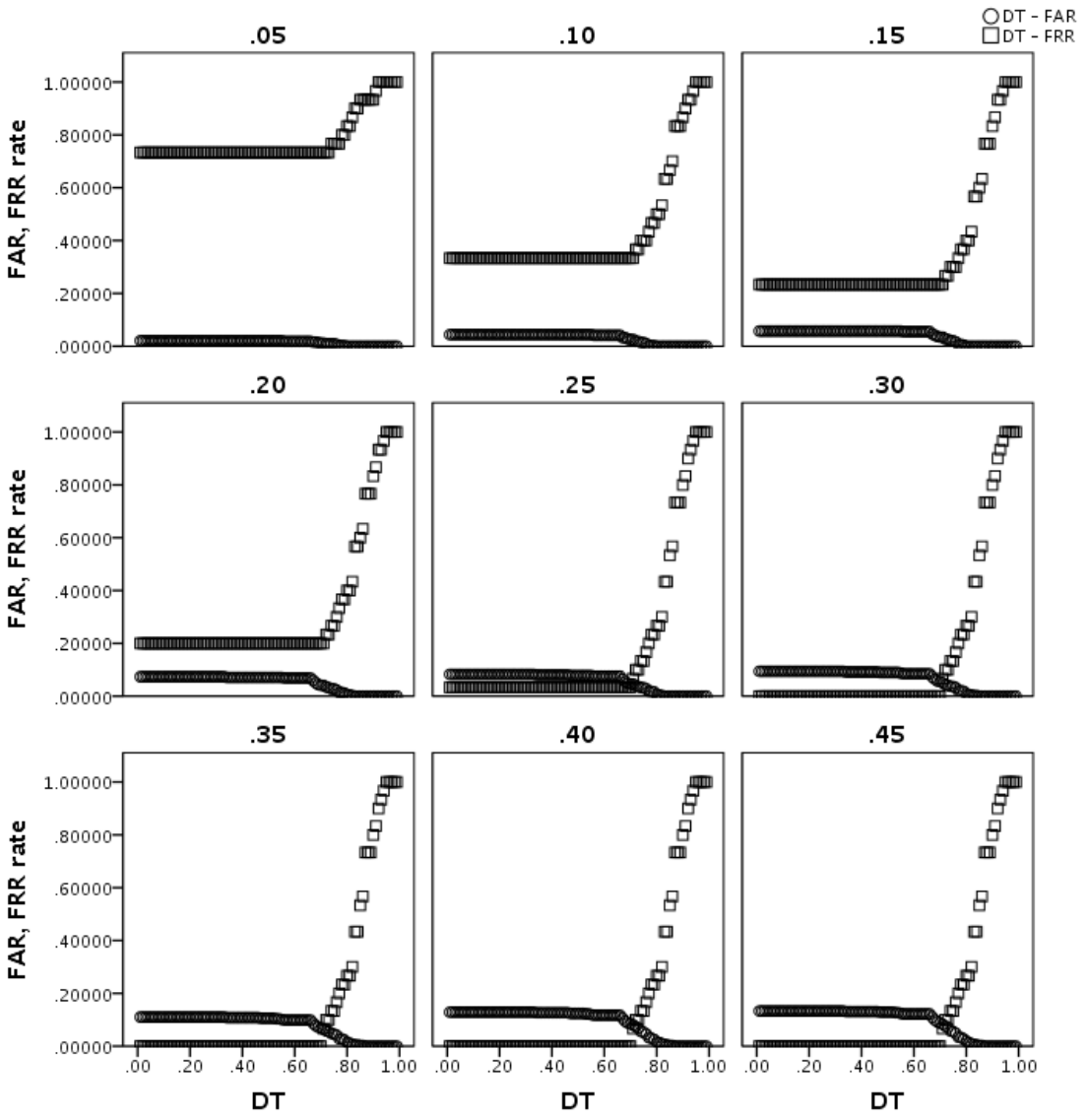

Figure 5: False acceptance and rejection rates as a function of the decision threshold, for 9 levels of allowed time variation between template and candidate input.

A careful look at the table indicates that setting ATV at .25 always resulted in at least 3.3% FRR, the same rate which was observed when the similarity metric was not put in place. Given that the FRR is calculated with only 30 comparisons, this means there was only a false rejection due to the total time variation control that could not be avoided.

Notice, however, that setting decision threshold to .7, it was possible to reduce the FAR from 8.92% to 4.6%. Similarly, for the .30 ATV level and the same .7 DT, the FAR was reduced from 9.43% to 5.52%. It is, therefore, clear that using the ComplHamming metric could indeed improve the resistance to impostor authentication attempts.

DT	ATV	FAR	FRR	DOT
0.7	0.3	5.52%	0.00%	5.52%
0.7	0.25	4.60%	3.33%	5.68%
0.69	0.3	5.75%	0.00%	5.75%
0.69	0.25	4.83%	3.33%	5.87%
0.68	0.3	6.44%	0.00%	6.44%
0.68	0.25	5.52%	3.33%	6.45%
0.7	0.35	6.90%	0.00%	6.90%
0.69	0.35	7.13%	0.00%	7.13%
0.67	0.25	6.44%	3.33%	7.25%
0.67	0.3	7.36%	0.00%	7.36%

Table 3: The 10 recognizer instances with lower DOT, out of 891 simulations using 9 levels of allowed time variation, 99 levels of decision threshold (as per table 1), with the sampling coefficient set to 1.

Computational Performance

The simulation tool was imported to an Android Test Project, using Android's testing framework, and 414315 consecutive authentication simulations were performed on a clean Samsung Galaxy mini device[34]. Recognizers were instantiated with all the combinations of the parameters in Table 1, except for the similarity metric, which was set to ComplHamming. Each operation took an average of 0.23 milliseconds, which suggests that the recognizer is indeed fast.

3.3.3 Discussion

The exploratory data analysis indicates that for the dataset were authentication simulations were run, the proposed algorithm is feasible. It's computational efficiency was attested by performing the actual operations that a Java implementation of the algorithm requires, on stock, low-end devices. Furthermore, by having the recognizer enforce all the proposed mechanisms and setting the decision threshold to around .7 and the allowed time variation to 0.25 or 0.30, the false acceptance and false rejection rates can be kept at low levels, in the region of 5%. There is evidence that all the steps of the matching algorithm do indeed improve accuracy.

Some recent fingerprint recognition techniques promise extraordinary low false acceptance and rejection rates. In the ongoing Fingerprint Recognition Competition[35], some algorithms which were evaluated to standardized benchmarks have achieved rates below 1%. However, these results aren't comparable to the ones here presented. Aside from the fact that achieving such precision in fingerprint recognition requires

34 A low-end device was chosen intentionally to approximate a worst-case scenario. Specifications: http://www.samsung.com/galaxyace/mini_techspec.html

35 FVC-onGoing: on-line evaluation of fingerprint recognition algorithms, https://biolab.csr.unibo.it/fvcongoing

technology that isn't yet available on smartphones (namely, precise optics), there is a fundamental difference between biometrics and knowledge-based systems as tap phrase authentication. Everyone has a different fingerprint. Not everyone has a different secret. The false acceptance rates in our analysis can be the result of users choosing the same (or very similar) tap phrases. It may be what as aptly been called a “password problem” (Maguire & Renaud 2012), not a recognition problem.

One limitation of this analysis is that, in a sense, the parameters found to be adequate to instantiate an accurate recognizer are, in fact, optimized for the dataset that was used. This is of special concern in the case of the false rejection rate, since for each recognizer instance only 30 genuine comparisons were performed, one for each pair of user template configuration and subsequent authentication attempt. One way to mitigate this is to perform subsequent evaluations with repeated authentication attempts, ideally over a long period of time (in order to also tackle memorability and skills-improvement effects, which were not addressed here). Designing such a study will benefit from the analysis that we presented, since solid and testable hypothesis can now be put in place.

3.4 Android Demo

As the oft-heard mantra goes, “simulations are doomed to success”. Not considering the practicalities of deploying authentication systems has been identified has one reason hindering adoption of post-password solutions (Dunphy et al. 2010). With the explicit objective to demonstrate the practical feasibility of our approach, we developed a small Android application in which tap authentication can be experienced.

The proof-of-concept application has only three features:

1. Configuration of a template: a three-step process similar to Android's pattern configuration, with an added tap phrase visualization facility.
2. Authentication: for subsequent tap phrase entry and matching.
3. Settings: an Android-standard preferences manager.

These features are shown in subsection 3.4.1. Designing interaction around the recognizer also raised the question of entry confirmation: how does a user tells the system that he has finished entering the tap phrase? The demo includes three options to address this problem – using buttons, gestures, or not requiring user confirmation at all. The trade-offs between these alternatives discussed in subsection 3.4.2. Deploying an actual application also had the collateral effect of producing a code base that developers can apply to tap phrase recognition problems in other contexts. The main re-usable components are shortly presented in section 3.4.3.

3.4.1 A Top-down View

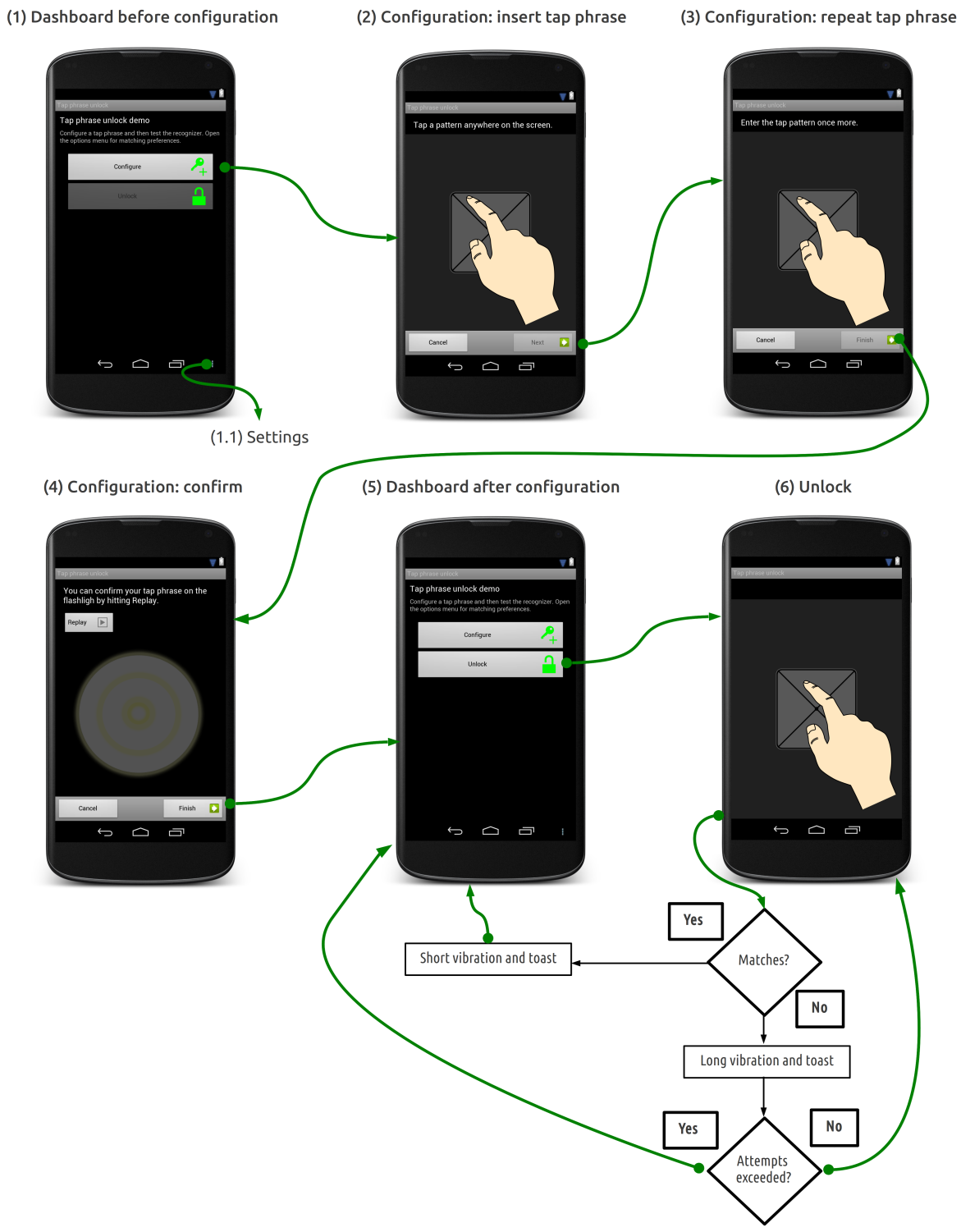

Figure 6: Digital wireframe for the Android demo application. Arrows represent the main transitions. Some non-GUI actions and transitions represented in flowchart style.

Figure 6 illustrates the interaction flow in the demo application. The application starts at the dashboard screen (1), which connects to the configuration and authentication screens. The second is blocked until a template is configured.

From the dashboard, using Android's option buttons, it is also possible to initiate a preferences screen (1.1). This screen defined declaratively in an XML file and rendered by Android depending on version and theme. The following preferences can be set:

- Decision threshold
- Allowed time variation
- Entry confirmation style (see subsection 3.4.2)
- Number of allowed attempts

Configuration is performed through three successive screens (2 to 4). The user is required to enter the tap phrase two times, as is customary. The template is recorded from the second example, and only if it is sufficiently similar to the first. In the third step, the user can visualize the tap phrase as a succession of flashes, and then confirm.

Once the configuration is completed, the “unlock” option is opened in the dashboard (5) and the user can then experience authenticating with a tap phrase (6). Successes and failures are acknowledged through short and long vibrations, respectively, and also Android's toasts (self-dismissing text notification). In the case of failures, the toast indicates how many attempts are left. Confirmation of input depends on the selected mode. The three available mechanisms are explained in the next section.

3.4.2 Entry Confirmation

When a user releases the screen, one of two things might be happening: either the pattern insertion was completed or a new off interval was started. To overcome this ambiguity, a number of approaches are possible.

The obvious approach, and the one we initially followed, is to set a timer after each release event and wait for another touch. If the timer ends without further touch events, one assumes that the interaction is over. This approach was abandoned because, in practice, it is ineffective, for two reasons. First, it forces “off” words to have an upper bound equal to the timer's length, thus limiting the richness of phrases that the users can select. Secondly, it imposes a wait period before an input is accepted or rejected, adding to the task completion time, which impacts usability. In effect, a trade-off between security and usability is created: if the timer length is too short, the task completion time improves but the theoretical key space is reduced, and vice-versa.

We therefore explored two other modes of operation, and decided to leave it to the developer which to make available to the user. Both approaches have advantages and disadvantages that we make clear. The specific use case should inform the choice of the appropriate mode.

Mode 1: Action-triggered Confirmation

In this interaction style, the user is required to perform an additional action after inserting the tap phrase. We propose two variations, one using buttons and another on-screen gestures.

Buttons: in the bottom of the screen, two buttons appear: “Clear” and “OK”. The first resets the input logger; the second triggers the recognizer.

Gestures: the user performs a swipe gesture[36] on the screen; left-to-right to confirm and right-to-left to clear. To capture the confirmation gesture, a standard recognizer is attached to the screen where the tap phrase is entered.

The main advantage of this style, in both variants, is the ability to correct errors using the “clear” action. Quick recovery from self-detected errors tends to have positive impact on user experience, and has very recently been identified as one of the reasons why users favor Android's pattern unlock over PINs, despite increased number of errors and task completion times (Zezschwitz et al. 2013). Conversely, usability is negatively impacted by having to perform an additional action. In the variant that resorts to buttons, finding them on the screen can also prevent inconspicuous interaction, at least for the novice user. For the gesture variant this is a lesser issue, since the user can swipe anywhere on the screen, using just one finger. However, since gesture detection can fail if the user does not perform the gesture within the detector's parameters, entry errors can be increased.

Mode 2: Continuous Verification

Another possible approach is to calculate similarity every time that a tap phrase candidate emerges, that is, every time the user releases the screen. This approach also resorts to a timer, but one that can be much longer, since it will only run out if the input does not match the template – if it matched, that last candidate already had been accepted.

In pseudo-code:

36 Gestures, Android Design Patterns, http://developer.android.com/design/patterns/gestures.html.

```
function onReleaseEvent()
   if similarity(template, candidate,...) > threshold
          accept();
   else
          startTimer();

function onTimerFinished()
   reject();
```

The great disadvantage of this approach is that it noticeably reduces the key space. For example, the input $\{200_{down}, 200_{up}, 200_{down}, 200_{up}, 200_{down}\}$ will be accepted if the template is either $\{200_{down}, 200_{up}, 200_{down}, 200_{up}, 200_{down}\}$, or $\{200_{down}, 200_{up}, 200_{down},\}$ or $\{200_{down}\}$. In other words, a triple tap covers the space of single, double and triple taps phrases.

Comparison

Table 4 summarizes the advantages and disadvantages of both modes of operation and respective variants. The observations reflect the general case. Users may miss the right button in the action-triggered mode, or be able to use the button without visual feedback, but these tend to be the extremes of novice and proficient users.

Mode of operation / variant	Provides error correction?	Increases entry errors?	Increases task completion time?	Requires visual feedback?	Reduces key space?
Action-triggered / buttons	Yes.	No.	Yes – finding the button and performing action.	Yes.	No.
Action-triggered / gestures	Yes.	Yes, due to gesture detection failures.	Yes – recalling the gesture and performing action.	No.	No.
Continuous	No.	No.	No.	No.	Yes.

Table 4: Advantages and disadvantages of the modes of operation

These factors, instead of imposing a trade-off, cross-cut both the usability and security dimensions. In particular, although continuous authentication reduces the key space, security considerations only make sense in the context of a threat model. In reference to the one presented in section 2.4.2, in which the adversary is a casual

observer in a social setting, to the extent that this mode of operation affords non-visual interaction and consumes less time than the alternative, it can be said to favor inconspicuous use. In this sense, security is improved, offsetting at least some of the effect of reducing the key space. A more conservative approach would be to use the action-triggered / gestures style, which also does not require visual feedback, but does not reduce the key space.

3.4.3 Utility Components

In the course of implementing the demo, a number of UI and support components were developed. To facilitate reuse, these components were packaged in an Android library project, which is available online[37].

The three main re-usable components are:

- *TapPhraseRecognizer*, which contains an implementation of the recognition algorithm, providing an API to match two tap phrases.
- *TapPhrasePad,* a UI component (an Android View) for entering tap phrases, as seen in screens 2, 3 and 6 of figure 6. It encapsulates a logger that keeps track of touch and release events over time. It also implements the plumbing behind the action-triggered modes of confirmation, and allows registering a listener for the continuous recognition mode.
- *TapReplayView,* a UI component that can replay a tap phrase in several output channels: visually, through a flashlight metaphor (as seen in screen 4 of figure 6), through sound, with a dial-like tone, and through haptics, using the vibration actuator.

More detailed considerations and implementation details are available in the the online documentation.

3.5 Summary and Outlook

This chapter presented a simple, efficient and accurate tap phrase recognizer designed for smartphone authentication. The recognizer algorithm and the deployment to the Android platform consummate the technical research objectives of this dissertation.

Tap phrase recognition was previously described in the literature. RythmLink (Lin et al. 2011) and TapSongs (Wobbrock 2009) are founded upon the same interaction style. Both, however, require more than one template to “train” the recognizer, which is

37 https://github.com/diogomarques/android-tap-phrase-detector

clearly an inconvenience to users. This inconvenience may result in the choice of poor tap phrases, or rich tap phrases that aren't ever changed. Our technique shows that this is not necessary: with a single example, we have observed false acceptance rates in the order of 5% (including the cases where users select the same tap phrase), while keeping false rejections also at minimal levels.

The fact that this recognizer was deployed to Android adds value to the contribution. Deployability issues have in the past plagued proposal for new authentication methods, and are at the core of much (meta-) debate. In fact, some authentication methods proposed in the literature were shown to have design flaws upon closer inspection (Dunphy et al. 2010; Perković et al. 2011; Tari et al. 2006; Maguire & Renaud 2012; Biddle et al. 2012). There seems to be a sense in the mobile HCISEC community that the platforms the industry provides are at a point where any advances are incremental. The upside is that smartphones are now evolved and mature enough so that, although research is meant to be forward-looking, there is a clear opportunity to close the time gap between knowledge-production and improving the lives of people. There is an unmistakable trend in major research outlets of deploying functioning prototypes, often to Android.

In section 3.3.3 we have identified some limitations which will be the target of continued efforts. Bigger data sets are needed for more robust accuracy assessment. Furthermore, interactions between accuracy and a larger context are in order. Can users remember tap phrases in continued usage? Can they remember more than one? Do they become more accurate by training? What other contextual and ecological factors in their daily lives can influence accuracy? Open questions like these will be the focus of future work.

There is also opportunity in exploring the recognizer, and software components implemented, for other domains. For instance, developers can use these components to program tap patterns into their systems by demonstration. The recognizer can be co-opted to add another category of gestures to the tool-belt of mobile interaction designers

Chapter 4

User Study: “Out in the Open” Authentication

This chapter presents the first of three user studies that were conducted to address two dimensions. First, the usability dimension, which is evaluated with the understanding that causing too much of an inconvenience creates an incentive for users to not secure their smartphones. Secondly, the security dimension, which is evaluated within the frame of a threat model in which the adversaries are not sophisticated hackers, but instead actors that act opportunistically, as defined in section 2.4.2.

In this first study, tap phrase authentication is compared to the two leading approaches, PIN and Android's pattern unlock, in an “out in the open” setting. The study includes two parallel experiments, one in which users performed unlocking tasks, and other where they performed simulated shoulder-surfing attacks. The disposition of participants emulated close-to-perfect observation conditions on the part of the attacker (see figure 7). This represents a worst-case scenario, but one that is not necessarily uncommon, if we consider that users may in fact perform authentication in settings like public transportation. It also compensates for the fact that the adversary only has one chance to shoulder-surf, whereas “in the wild” someone familiar to the user may have many opportunities to do so.

The usability dimension is, as previously stated, understood more generally than “user performance”. One aspect of usability that falls outside user performance is user experience (UX), that is, “perceptions and responses that result from the use or anticipated use of a system”[38]. These perceptions can influence the acceptance of new technologies. This is an aspect that is explored in this study more thoroughly than in the following two. The approach that was taken to UX evaluation was a quantitative one, namely using a well validated questionnaire.

38 ISO 9241-210:2010 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, available at http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=52075.

Figure 7: Subjects in the "out in the open" condition.

4.1 Research Questions

This study aims at establishing a baseline for the understanding of PIN, Android pattern and tap phrase unlock by answering the following questions:

1. Do the three evaluated methods provide similar usability?
2. Under good observation condition, do the three methods offer similar resilience to shoulder-surfing?

4.2 Methodology

For the experimental part of this study, a repeated measures design was employed. There were two parallel moderated experiments. In one, users performed authentication tasks using the three methods. In the other, they observed another user authenticating and tried to repeat their secret code. In both, the independent variable was the unlock method, with three levels: draw pattern, PIN and tap phrase.

After completion of experimental tasks, users responded to the UX questionnaire. This questionnaire was a version of the AttrakDiff instrument first proposed in Hassenzahl et al. (2004). Choosing standardized over ad hoc questionnaires has a clear advantage: the latter have already gone through psychometric qualification; and in the case of AttrakDiff, longer term reviews of its application are already available (Bargas-Avila & Hornbæk 2011). To keep the evaluation procedure short, the 10-item version (van Schaik et al. 2012) was chosen.

4.2.1 Apparatus

For the experiments, a single Google Galaxy Nexus device with Android 4.1 was used. An application for data-gathering was developed, implementing the three unlocking methods: 1) Android's graphic (draw) pattern, 2) PIN and 3) tap phrase. For the tap unlock method, we used the recognizer described in the previous chapter; for PIN, we created a simple form with a text field and had the system present a numeric keyboard; finally, for draw pattern unlock, since Android is open-source, we extracted the code actually deployed to commercial devices.

Between tasks, the Android application also prompted the user to answer the single ease question (SEQ) and gather results. The SEQ is a standardized usability instrument proposed in Sauro & Dumas (2009), whereby users are asked to complete the statement "Overall, this task was:". We employed the recommended seven point scale, where 7 is "very easy", and 1 "very difficult".

The application generated an XML file containing logs of every user interaction, including the answer to the SEQ answers.

The post-experiment questionnaire was administered with a Google Documents web form. Users were asked to answer it immediately after the experiments. It started with standard demographic questions and then proceeded to the 10 semantic differentials for UX assessment. Each semantic differential had a seven-point scale, with 1 being the negative adjective, and 7 the positive. Answers were collected automatically to a spreadsheet.

4.2.2 Participants

Thirty volunteers were recruited through mailing lists, social media and word-of-mouth. All were students or research staff. Seventeen were male and thirteen female. Ages ranged from 21 to 50 years old, with the average being 26 (SD = 6). Only two participants reported not being at all familiar with smartphones; 13 reported being extremely familiar. Eleven participants reported currently using either a PIN or password to unlock their personal devices; 8 reported using Android's draw pattern. Participants were offered no compensation.

4.2.3 Procedure

Participants were introduced to the experiments and explained their roles as unsuspecting user and opportunistic observer. They were given no mention that one of the unlock methods was proposed by the researchers. The participant playing unsuspecting user was given a smartphone, with the test application already launched.

For each unlock method, users learned or configured their code and tried it out in seclusion until they were confident that it was memorized.

The observer, which we referred to as challenger, was then called to shoulder-surf while the other performed the authentication task, having a maximum of 10 trials to complete it. Upon completion, the application prompted the user to answer the SEQ. The device was then given to the challenger, which also had 10 trials to replicate the code. Unlocking methods were presented in random order. For each method, a participant acted one time as the unsuspecting user and one time as the opportunistic observer. After finishing the experiments, participants were directed to respond the online questionnaire.

Experimental sessions were conducted in various locations around the university campus, including meeting rooms, offices, bars and sidewalks, as per participant convenience. We reasoned that although this may introduce greater variability in measurement, it increases ecological validity. That is, the experiment mimics as much as possible real-life situations.

Random 4-digit PINs were supplied to the user by the application; the length 4 being chosen because it is widely used, as the default option in ATMs, SIM cards, etc. For draw patterns, the application also generated random 5-point patterns; the length 5 being the median of a small-scale (N = 11) survey of colleagues. In pilot runs, users showed great difficulty in replicating random tap phrases. We therefore ended up letting them configure their own, with the limitation that it had to contain at least three touches. We tried to limit biases in memorability by allowing unbounded learning time for PINs and draw patterns. Providing random codes for these two methods may lead to greater resilience to attacks, in comparison to tapping, which, being user-configured, may be more intelligible. Tap patterns, not being bounded in time, also can increase task completion times. These limitations are reasonable in so much as they favor the alternatives against which tap authentication is evaluated.

4.2.4 Measures

In the unlocking experiment, we acquired the total time it took to complete the task and the SEQ score. The task completion time was measured from the moment the user was presented with the screen to the moment when authentication was successful. This measure therefore also encapsulates:

- the time users take to situate themselves before starting input; and
- errors in input and recovery from them.

In the observer experiment, we measured the success within 10 trials. The higher this success rate is, the lower the method's resilience to shoulder-surfing is.

Table 5 shows the semantic differentials in the UX questionnaire. We summarized a pragmatic quality score for each user by calculating the average rating of the first 4 answers. We did the same to synthesize and hedonic quality score, using the following 4 answers. The beauty and goodness quality score are simply the rating given in the last two answers, respectively.

	I judge the unlocking method to be
PQ1	Confusing – Structured
PQ2	Impractical – Practical
PQ3	Unpredictable – Predictable
PQ4	Complicated – Simple
HQ1	Dull – Captivating
HQ2	Tacky – Stylish
HQ3	Cheap – Premium
HQ4	Unimaginative – Creative
	I judge the unlocking method overall to be
B1	Ugly – Beautiful
G1	Bad – Good

Table 5: Semantic differentials in the user experience questionnaire. From van Schaik et al. (2012).

The task completion times did not follow a normal distribution, as indicated by a Shapiro-Wilk test. Friedman tests were therefore employed, as they were for the ordinal data obtained in questionnaires. For the challenge task, since outcomes are binary, a Cochran's Q test was used. The alpha level was set at 0.05. For post-hoc tests, it was adjusted with the Bonferroni correction.

4.3 Results

Table 6 shows a summary of statistically significant effects in pairwise analysis, which were only assessed when there was a significant omnibus effect of method on measure. The following subsections give descriptive statistics and the results of hypothesis testing for each measure.

Measure	Pair	Point estimate
Unlock task completion time	Draw / PIN	3.82s / 2.81s
Perceived unlock task ease	Tap / PIN Tap / Draw	6 / 7 6 / 7
Perceived pragmatic quality	PIN / Draw Draw / Tap Tap / PIN	5.88 / 5.05 5.05 / 3.91 3.91 / 5.88
Perceived hedonic quality	Draw / PIN PIN / Tap	5.30 / 3.23 3.23 / 4.57
Perceived beauty	PIN / Draw Draw / Tap Tap / PIN	3.09 / 5.37 5.37 / 4.87 4.87 / 3.09
Perceived goodness	-	-
Shoulder-surfing attack success	-	-

Table 6: Summary of significant effects in follow-up analysis.

4.3.1 Unlock Task Completion Time

The mean task completion times were 2.81s (SD = 1.56s) for draw pattern unlock; 3.82s (SD = 2.60s) for PIN unlock; and 3.73s (SD = 3.10s) for tap pattern unlock (see figure 8). The effect of method on this metric was significant ($\chi^2(2) = 17.267$, p =.000). Pairwise, significance was only found between draw pattern and PIN unlock (Z = -2.437, p = .015, r = .315). There was no evidence of a significant effect between tap pattern unlock and both PIN (Z = -.627, p = .530, r = .081) and draw pattern unlock (Z = -2.273, p = .023, r = .293). We conclude that using a draw pattern was faster than using PIN; in the remaining comparisons, neither method showed to be less or more time-consuming than the other.

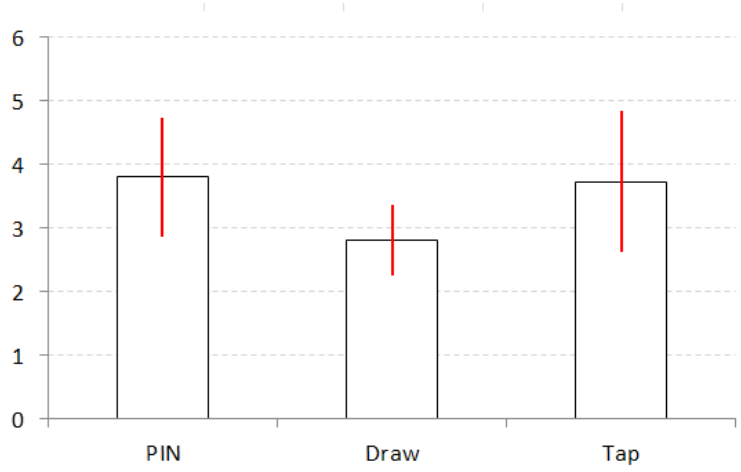

Figure 8. Mean task completion time (in seconds) for each method. Error bars indicate the 95% confidence interval.

4.3.2 Perceived Unlock Task Ease

The median SEQ score was 7 (IQR = 1) for draw pattern unlock; 7 (IQR = 1) for PIN unlock; and 6 (IQR = 3) for tap pattern unlock (see figure 9), and statistical significance was found ($\chi^2(2) = 7.750$, $p = .021$). Pairwise, tests showed no significant differences between draw pattern and PIN unlock ($Z = -.690$, $p = .490$, $r = 0.089$), and significant differences between tap pattern unlock and both PIN ($Z = -2.670$, $p = .008$, $r = .345$) and draw pattern unlock ($Z = -2.864$, $p = .004$, $r = .370$). We conclude that unlocking was perceived as most difficult when using the tap pattern method. Draw pattern and PIN unlocking were perceived as easiest, with no significant difference between them being found.

4.3.3 Perceived Pragmatic Quality

The mean perceived pragmatic quality (PQ) score was 5.05 (SD = 1.41) for draw unlock, 5.88 (SD = .69) for PIN unlock, and 3.91 (SD = 1.65) for tap pattern unlock (see figure 10). Again the effect of method on this metric was significant ($\chi^2(2) = 24.748$, $p = .000$). Pairwise tests indicate that the effect is significant for all pairs (PIN - draw pattern: $Z = -2.490$, $p = .013$, $r = .321$; tap pattern - draw pattern: $Z = -3.441$, $p = .003$, $r = .444$; tap pattern - PIN: $Z = -4.146$, $p = .000$, $r = .535$). We conclude that PIN unlock was perceived has having superior pragmatic quality, followed by draw pattern unlocking, and then tap pattern unlocking.

4.3.4 Perceived Hedonic Quality

The mean perceived hedonic quality (HQ) score was 5.30 (SD = .99) for draw unlock, 3.23 (SD = .97) for PIN unlock, and 4.57 (SD = 1.47) for tap pattern unlock (see

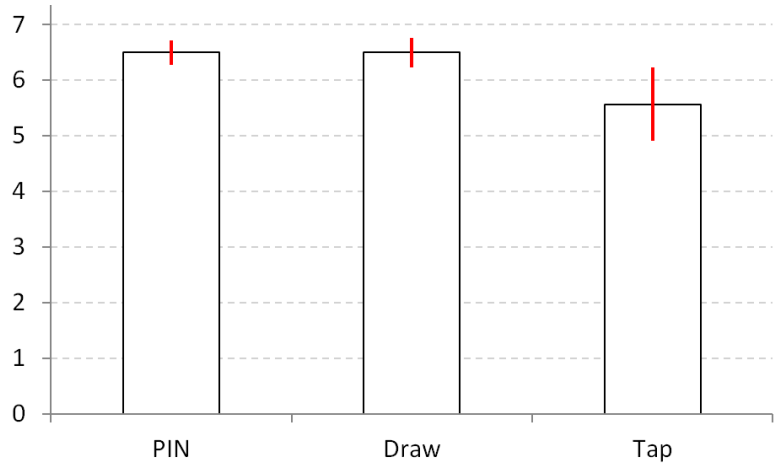

Figure 9: Mean single ease question score in a 1 (very difficult) to 7 (very easy) scale. Error bars indicate the 95% confidence interval.

figure 10). A significant main effect was found ($\chi^2(2) = 30.360$, $p = .000$). Pairwise, significant effects are present for the pairs draw pattern – PIN ($Z = -4.789$, $p = .000$, $r = .618$) and tap pattern - PIN ($Z = -4.306$, $p = .000$, $r = .556$). However, no significant effect was found for the tap pattern – draw pattern pair ($Z = -2.364$, $p = .018$, $r = .305$). We conclude that PIN unlock was perceived has having the lowest hedonic quality, and that, in this regard, a difference between draw and tap pattern methods could not be established.

4.3.5 Perceived Beauty

The mean perceived beauty score was 5.37 (SD = .83) for draw unlock, 3.09 (SD = 1.10) for PIN unlock, and 4.87 (SD = 1.37) for tap pattern unlock (see figure 10). Statistical significance was again found ($\chi^2(2) = 41.397$, $p = .000$). Post-hoc analysis indicates that the effect is significant for all pairs (draw pattern - PIN: $Z = -4.227$, $p = .000$, $r = .546$; tap pattern - PIN: $Z = -3.429$, $p = .001$, $r = .443$; tap pattern – draw pattern: $Z = -2.498$, $p = .012$, $r = .322$). We conclude that draw pattern unlock was perceived has having superior beauty, followed by tap pattern unlocking, and then PIN unlocking.

4.3.6 Perceived Goodness

The mean perceived goodness score was 5.17 (SD = 1.37) for draw unlock, 4.97 (SD = 1.30) for PIN unlock, and 4.93 (SD = 1.63) for tap pattern unlock (see figure 10). A Friedman test was ran, and no statistical significant effect was found ($\chi^2(2) = 1.200$, $p = .549$).

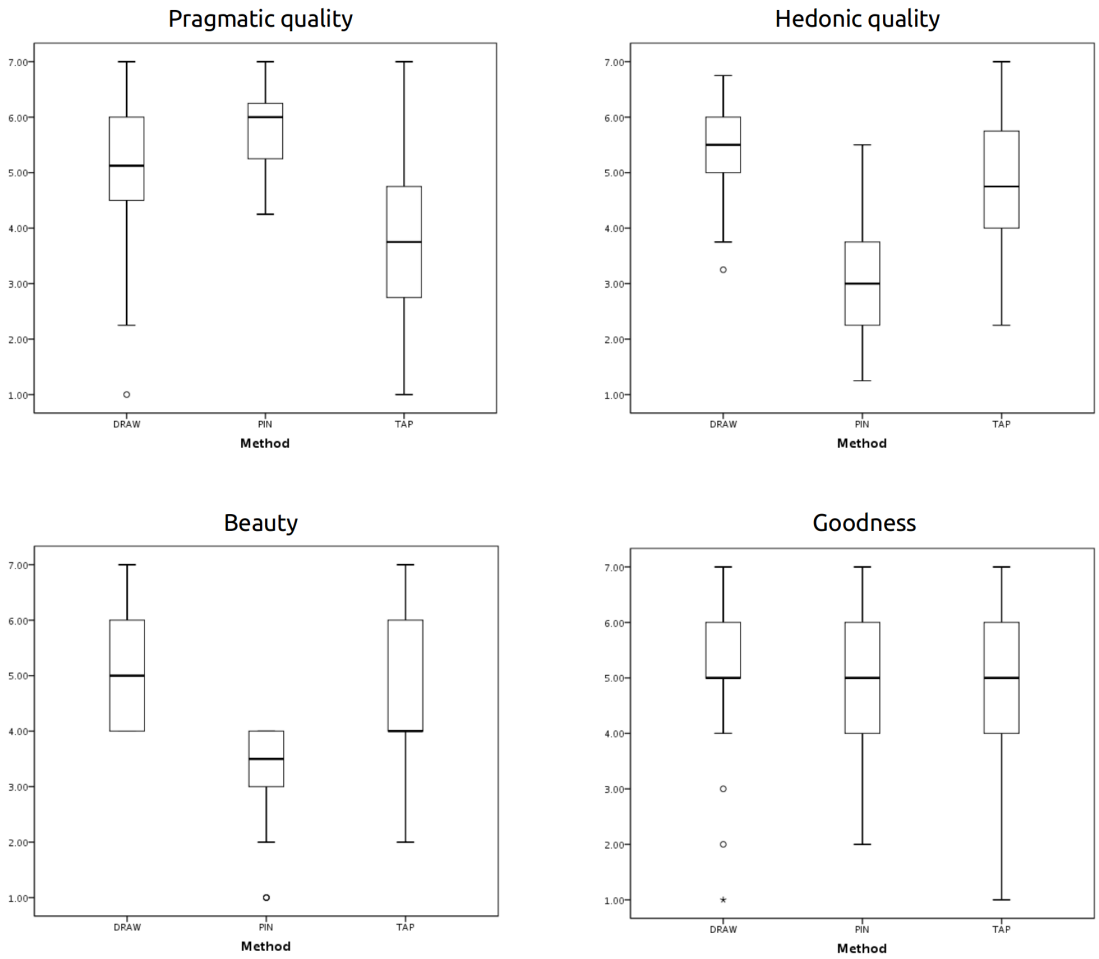

Figure 10: Distribution of ratings from the UX questionnaire, where 1 is the closest to a negative connotation and 7 the closest to a positive one.

4.3.7 Shoulder-surfing Attack Success

In the shoulder-surfing task, subjects were able to successfully replicate another person's code (within 10 trials) 5 out of 30 times when using either draw or tap pattern unlock, and 9 out of 30 times when using PIN unlock. To see if the unlock method had a significant effect on this task's completion rate, we ran a Cochran's Q test, and no statistical significance was found ($\chi^2(2) = 2.462$, $p = .292$). In conclusion, no evidence of method having an effect on resilience to shoulder-surfing attacks was found.

4.4 Discussion

This first user study indicates that tap phrase unlocking is comparable to the two leading methods in terms of usability and resilience to shoulder-surfing.

As for user performance, there is no statistical evidence that tap phrase unlock is more time consuming that either PIN or draw unlock. The subjective perceptions were

more mixed. Tap unlock was perceived to be more difficult, but still easy, with median score being 6 where 7 means “very easy”. It was also found to have lower perceived pragmatic quality than the alternatives. It had, however, better hedonic quality and beauty ratings than PIN unlock.

It is easy to see why PIN unlock can be perceived as easier: we are accustomed to them. PIN's are used in many critical contexts, giving them some measure of respectability, which may be influencing the pragmatic quality ratings. But for the same reasons, PINs can be seen as dull, hence being worse than tapping – and, indeed, the worst – in hedonic quality and beauty ratings. Draw pattern unlock presents the highest beauty score, but still no differences in hedonic quality were found in relation to tap unlock.

This study also addresses shoulder-surfing resilience. The results showed no statistical significance, but the fact that PINs were successfully replicated by the subject playing the attacker 9 times, in comparison to the 5 observed in the tap and draw pattern unlock methods, should give us pause. This is likely a symptom of a phenomenon we observed while doing the experiments: 4-digit PINs are fast to memorize, at least for a short period of time. Committing a tap or draw pattern to memory takes more time. While the user that was authenticating had unlimited time to memorize the code, the attacker could only do it in the short period she was observing the victim. Anecdotal evidence of this is present in the logs. When users were given a random PIN and prompted to try it, in almost every case they only tried once before signaling that they had learned it. For draw patterns, there are many instances where users tried the code they were given up to 3 times before indicating it was memorized.

In summary, regarding the question of whether the methods provide similar usability, there is evidence that this is the case, although the subjective perceptions of tap phrase authentication are mixed. Regarding shoulder-surfing resilience, under perfect observation conditions, we found evidence that all methods are very susceptible.

Chapter 5

User Study: Inconspicuous Authentication

The threat model that tap phrase authentication aims to address is one where the social context matters. Using an input modality that does not require the visual channel has the potential to allow inconspicuous behavior, which users can leverage for self-protection. The first user study establishes a dreading baseline: when observation is possible, shoulder-surfing attacks are very much feasible against the assessed authentication methods. This second study explores the feasibility of authenticating away from prying eyes.

This study consisted of a single experiment, where subjects were asked to configure/learn a code, and then unlock the device under a table (see figure 12). Again, tap unlock was compared to the leading approaches: PIN and Android's pattern unlock.

Having the users authenticate under the table was a way to isolate the effect of not having visual feedback or an observation angle for a third-party, while maintaining a realistic scenario for inconspicuous interaction with a smartphone. The user study in the next chapter gives further insight into strategies for dissimulated interaction.

5.1 Research Questions

This study addresses the following questions:

1. Do the three methods provide similar usability when there is no visual feedback?
2. For each method, how is usability impacted by unlocking being performed inconspicuously, in comparison to the previous setting?

5.2 Methodology

Nineteen out of the 30 subjects that participated in the previous study were again recruited. The same apparatus (device with data-gathering app) was used. The 19 subjects that also participated in this study averaged 27 years of age (SD = 7, range: 21-50). The procedure was similar to the one used in the previous unlock experiments,

Figure 11. Subject in the "under the table" condition.

except for the placement of device away from sight. The observer experiment does not apply. Users were only allowed to look at the screen between trials, thus observing if they were successful or not, and repositioning themselves for a new trial in the latter case. One additional measure was gathered from the logs: the number of input errors. This measure was extracted in this case because there was a reasonable expectation, from pilot testing, that errors could vary considerably depending on the authentication method.

5.3 Results

In the following subsections, statistics for both conditions are constrained to the 19 subjects that participated in this study. Therefore, small changes in metrics for the visual feedback condition are to be expected.

5.3.1 Unlock Task Completion Time

In the condition where visual feedback was available, the mean task completion times were 2.77s (SD = 1.31s) for draw pattern unlock, 4.34s (SD = 3.15s) for PIN unlock, and 4.14s (SD = 3.84s) for tap pattern unlock (Figure 12). Without visual feedback, the mean task completion times were 30.32s (SD = 31.42s) for draw pattern unlock, 43.14s (SD = 32.52s) for PIN unlock, and 6.18s (SD = 9.21s) for tap pattern unlock.

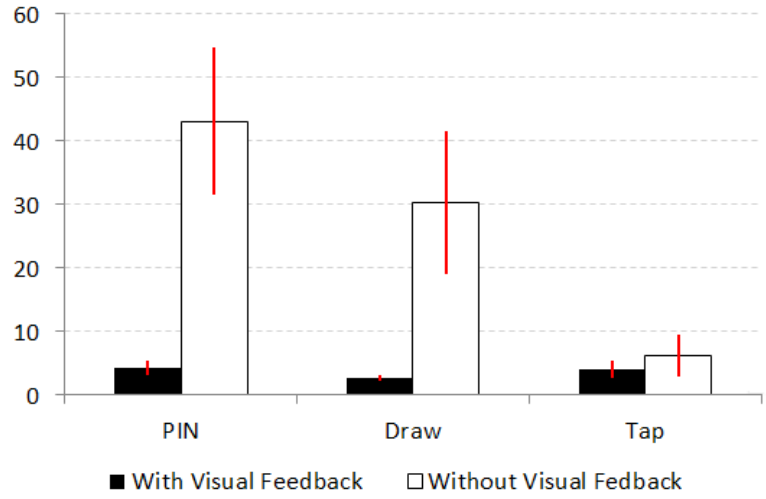

Figure 12. Mean task completion times (in seconds) for each method and each visual condition. Error bars indicate the 95% confidence interval.

The unlock method had an effect on the time it took to complete the task without visual feedback ($\chi^2(2) = 15.474$, $p = .000$). Pairwise comparisons were conducted, and no significant differences were found between PIN and draw pattern unlock ($Z = -1.690$, $p = .091$, $r = .274$). However, between tap pattern and both PIN ($Z = -3.380$, $p = .001$, $r = .548$) and draw pattern unlock ($Z = -2.978$, $p = .003$, $r = .483$) the effect was significant.

Comparing the visual to the non-visual condition for each method, there was no evidence of an effect for tap pattern unlock ($Z = -1.207$, $p = .227$, $r = .196$) There were, however, effects for both draw pattern ($Z = -3.783$, $p = .000$, $r = .614$) and PIN unlock ($Z = -3.823$, $p = .000$, $r = .620$).

We conclude that unlocking without visual feedback was significantly faster using a tap pattern than using a PIN or draw pattern. In this condition, a difference between PIN and draw pattern unlocking could be established. Furthermore, the latter two methods consumed significantly more time when there was no visual feedback in comparison to the previous setting.

5.3.2 Unlock Input Errors

When there was no visual feedback, the mean number of input errors was 2.84 ($SD = 2.544$) for draw pattern unlock, 3.53 ($SD = 3.325$) for PIN unlock, and .42 ($SD = 1.387$) for tap pattern unlock. Statistical tests showed that the differences were significant ($\chi^2(2) = 15.474$, $p = .000$). Pairwise, differences between PIN and draw pattern unlock were non-significant ($Z = -0.514$, $p = .607$, $r = .083$). However, again there were significant differences between tap pattern and both between PIN ($Z =$

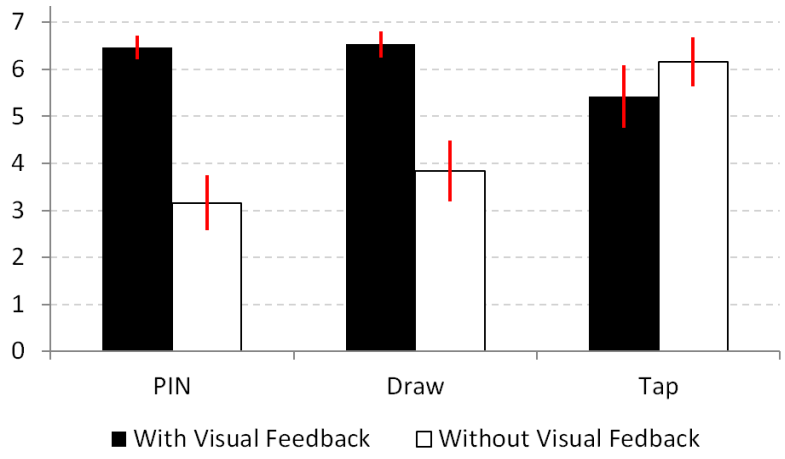

Figure 13: Mean single ease question score for each metho and each condition, in a 1 (very difficult) to 7 (very easy) scale. Error bars indicate the 95% confidence interval.

-2.894, p = .004, r = .469) and draw pattern unlock (Z = -3.020, p = .003, r = .490). We conclude that unlocking without visual feedback was less error-prone when using a tap pattern.

5.3.3 Perceived Unlock Task Ease

We again measured the perceived ease of completing the task using the SEQ with equally labeled levels (see figure 13). With visual feedback, the median score was 7 (IQR = 1) for draw pattern unlock, 7 (IQR = 1) for PIN unlock, and 6 (IQR = 3) for tap pattern unlock. Without visual feedback, the median score was 3 (IQR = 2) for draw pattern unlock, 3 (IQR = 3) for PIN unlock, and 7 (IQR = 1) for tap unlock.

When visual feedback was not available, the unlock method had an effect on the subject's perceived ease ($\chi^2(2) = 22.377$, p =.000). Post-hoc analysis yet again does not show significant differences between draw pattern and PIN unlock (Z = -1.361, p = . 174, r = 0.221), and shows them between tap pattern unlock and both PIN (Z = -3.613, p = .0000, r = .586) draw pattern unlock (Z = -3.536, p = .004, r = .574).

The pairwise comparisons between the visual and non-visual settings do not reveal a significant effect in the case of tap unlock (Z = -1.342, p = .179, r = .218). For the other two methods, such effects were present (Z = -3.454, p = .001, r = .560) and PIN unlock (Z = -3.742, p = .000, r = .607).

We conclude that, without visual feedback, tap unlocking was also perceived as least difficult. Comparing visual and non-visual entry, for tap unlock there was no evidence of a difference, contrary to draw pattern and PIN unlock, which were

perceived as more difficult in comparison to the condition where visual feedback was available.

5.4 Discussion

This study largely confirmed that tap phrase unlocking is an adequate solution for situations where a user's visual channel is not available. For PIN and draw pattern unlock the time it takes and the number of errors greatly increases in this condition. The same cannot be said for tap pattern unlock. The subjective perception of easiness, not surprisingly, is in line with these findings.

This is in stark contrast to the findings of the previous user study. Tap phrase unlocking may have comparable or even slightly worse usability when the visual channel is available. But when concealing authentication from one's sight and from prying eyes, there is clear evidence that the method offers considerable more usability than the alternatives.

Chapter 6

User Study: Tap Authentication for Blind People

To blind users, using touchscreen-based devices like smartphones is a challenge. As Guerreiro et al. (2008) observes, “most interactions […] require hand-eye coordination, making it difficult to for blind users to interact with mobile devices and execute tasks”. Nevertheless, security is not less important for the blind. In fact, blind users may be more exposed to observation, given the absent visual perception of surroundings.

Currently, the only widely available authentication method for blind users resorts to PINs and a screen reader. For instance, with the iPhone's VoiceOver facility, as the user passes its fingers through the screen, a voice reads out each key. A second touch is required to select. Azenkot et al. (2012) found that even experienced users took an average of 7.52s to authenticate themselves in this way. Moreover, not only is shoulder-surfing possible, but a vector for aural eavesdropping is also opened when the user is not wearing a head-set.

This third user study is an exploration into using tap phrase authentication as an inclusive – not adapted – technical solution. It addresses both the usability of tap phrase authentication in this specific population, and the affordance of the method to inconspicuous interaction. The 16 participants were asked to perform a tap authentication task and then come up with strategies for dissimulating interaction. This gave further insight into how users can easily adapt and self-protect in a threatening environment if the appropriate tools are provided.

6.1 Research Objectives

The objectives of this study were to understand if, after a short learning period, blind users could:

1. Perform authentication easily and in a reasonable amount of time, and;
2. Devise strategies for inconspicuous authentication.

6.2 Methodology

6.2.1 Apparatus

A single Samsung Galaxy mini smartphone, with Android 2.3, was used for the authentication task and subsequent role-playing procedure. The data-gathering Android application was modified to only include the tap unlocking method. A training mode was available in which data was not recorded and optional sound output (emission of a tone while the screen was being touched) was available. A short vibration was emitted on successful unlock.

Paper questionnaires were employed to gather demographic data, register responses to the single ease question, and record concealment strategies suggested by participants.

6.2.2 Participants

The 16 participants were volunteers recruited in a local vocational training institution for blind people. Two participants had some residual vision. Ages varied between 26 and 64 years old, the average being 47 (SD = 12). Twelve participants were male and 4 female. Although all participants had mobile phones, they reported having none or very little experience with touch-screen devices. Eleven reported being very familiar with using PINs in electronic devices, albeit in physical keyboards. Participants reported never or rarely using headphones paired with their mobile devices.

6.2.3 Procedure

Participants were initially introduced to the concepts and explained the tasks they were asked to perform. At this stage, they were given no mention that the tap unlocking was the method being proposed by the researchers. They were handed a device to feel and get accustomed to while being administered a short demographic questionnaire.

In a first stage, a moderated training session lasting approximately 5 minutes was conducted, in the following steps:

1. Users freely explored the touch-screen area with their fingers. When they touched any point in the screen, an audio tone was emitted. Participants were explained that the whole screen acted as a single button and were guided to explore the fact that tap phrases are composed of taps and breaks lasting in time.
2. Users were asked to imagine tap phrases that they could record and later use for unlocking. They experimented freely, with sound enabled, until they were confident that they had grasped the concept.

3. Users were introduced to the vibrotactile feedback emitted on authentication success (short) and failure (long).
4. With sound output now removed, users conducted a complete dry-run, first configuring a template of their choice, then attempting to unlock.

After training, participants were asked to again configure a template and then try to unlock. This time, the interaction was recorded in log files. Immediately after completing this task, users responded to the single ease question.

In the second stage of this study, participants were introduced to the shoulder-surfing threat and asked to imagine strategies they could use to conceal the input from potential observers. To facilitate this process, participants engaged in role-playing two scenarios: a meeting and a commute in public transportation. To that end, they were given a smartphone so they could simulate authentication. A facilitator gravitated at times around the participant to make him aware of possible visual observation angles. In the end, participants were asked to summarize the viable strategies they had identified.

6.2.4 Measures

For the unlocking task, we acquired:

1. The time it took to complete authentication, measured from the moment a facilitator clicked a start button and initiated the unlocking screen to the moment an input was accepted as the correct secret code;
2. The number of input errors, and;
3. The SEQ rating, from 1 to 7.

For the elicitation part of this study, the strategies indicated by participants were recorded in paper and occurrences counted. Since the alternatives mentioned were clear and not very numerous, no special categorization was performed.

6.3 Results

After training, all users were able to authenticate in the first trial, so there were no input errors to record.

The mean task completion time was 4.32s, with standard deviation 2.1s (see figure 14, left). A Shapiro-Wilk test indicated that the data is normally distributed (S-W = .890, df = 16, p = .056). A one-sample t test was conducted at an alpha level of .05 to evaluate if tap unlocking was faster than the 7.52s benchmark for PIN with VoiceOver found by Azenkot et al. (2012). The test showed significance (t = -6.062, df = 15, p = .

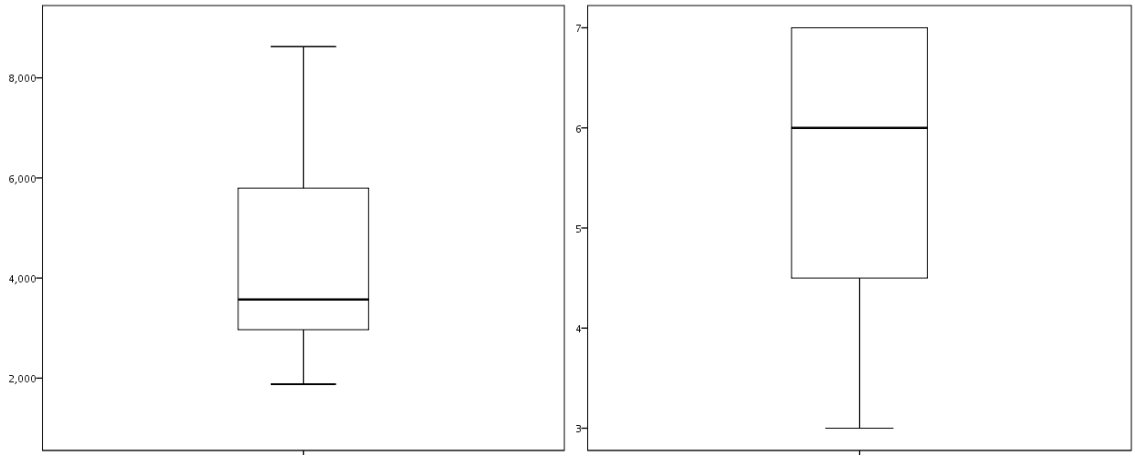

Figure 14: Task completions time (left) in seconds and perceived task ease (right) in a 1 (very difficult) to 7 (very easy) scale.

000, Cohen's d = -1.52), indicating that unlocking with taps was indeed faster than with PIN/VoiceOver.

The median SEQ score was 6 (IQR = 3), where 7 means "very easy" and 1 "very difficult" (figure 14, right), indicating that participants perceived tap unlocking as being easy to perform.

In the second stage of the study, inconspicuous authentication strategies were elicited through role-playing. The user-suggested approaches are summarized in table 7. Each user contributed, on average, 3 strategies (SD = 1). The top suggestions, with 9 occurrences, were performing authentication under the table or inside a pocket.

Strategy	**Occurrences**
Under the table	9
Inside pocket	9
Occluded by clothes (e.g. jacket)	7
Cover with one hand	5
Lean device against body	3
Inside bag / purse	3
Using device upside down	3
Move to an isolated location	2
Under the seat	1
Postpone2	1

Table 7: Suggested authentication concealment strategies. The left column identifies the strategy; the right column indicates how many participants suggested it.

6.4 Discussion

The results for task completion time and perceived easiness of authenticating with tap phrases are encouraging. Even so, the relatively large standard deviation in task completion time deserves a closer look. From our observations, there are two possible explanations for this fact: 1) some users, lacking the confidence and experience using smartphones, operated the device with an unusual level of caution, thus taking more time and 2) there may be, in fact, an extended initial period where a blind user needs to situate himself before starting tapping with confidence.

The top suggestions for inconspicuous authentication strategies include many cases that are made possible, or at least easier, by the tap phrase method. This is true not only for blind users, but in any situations where the visual channel is not available. For instance, the previous study already showed that PIN and Android's pattern unlock are much less usable when authenticating under the table, which was among most frequent strategies identified. The feasibility of actually using some of the selected strategies must, however, be further evaluated in realistic settings. For example, using a pocket may not be possible because hand movements can be constrained.

In conclusion, the research objectives were achieved. Blind users could authenticate easily and in lesser time than the most common method available in smartphones. They were also able to easily devise strategies for inconspicuous authentication, showing that with the right tools it is possible to self-protect against the threat of shoulder-surfing in social contexts.

Chapter 7

Conclusion

Mobile devices are becoming extensions of ourselves, permeating many aspects of our lives. Smartphones in particular have become more than personal assistants. They are, in many ways, intimate computers. For all the benefits that we can gain from this relationship, we also face new dangers. How can we trust a friend that puts us at risk every time we engage with it? We have found ways to cope with this same problem in interpersonal relationships. In the presence of others, we whisper. We wink. We nod. We pull closer. But it is still challenging to limit the exposure of interactions with our smartphones.

7.1 Summary

This document presents an authentication method that allows inconspicuous interaction, using tap phrases as passwords. It offers users more control on how they perform perhaps the most critical recurring interaction with a smartphone: establishing identity. By using tap phrases as shared secrets, users can choose to authenticate themselves overtly; but if they feel they might be exposing a secret to bystanders, they can do it away from sight.

The first contribution of this work is a novel tap phrase recognizer that was shown to be accurate and efficient. This recognizer was specifically designed for the authentication scenario, although it can be appropriated for other purposes. Our approach improves on previous work by requiring a single example for configuration, as is the norm with other varieties of knowledge-based authentication. Having to insert several examples to train a matching algorithm would have created incentives for users to select poor tap phrases and not change them frequently. This contribution was further enriched with the development of a proof-of-concept Android application. The first two research objectives are thus fulfilled respecting the principle of “designing with the adoption process in mind”.

The second contribution is an evaluation of the usability and security of this method. The first user study indicates that this approach is usable when compared to the leading unlock authentication methods, PIN and Android's pattern unlock. We found that the three alternatives are susceptible to shoulder-surfing when clear observation is possible. The second study validates that, unlike the alternatives, tap phrase authentication allows inconspicuous interaction, thus not only offering increased security (in relation to the defined threat model) but also enabling compliance with social norms. The third user study expands the understanding of usability to include accessibility. It indicates that the proposed method is inclusive and more usable by blind people than the typical PIN coupled with a screen reader.

7.2 Limitations

Computer security is in many ways like a short blanket. When we snuggle, our feet are left in the cold. Authentication through tap phrases reduces the threat of shoulder-surfing, and thwarts the smudge-attacks that touchscreens enable. This is valuable in the sense that it addresses an important threat model, where the adversary is not necessarily a security expert, but someone that has the incentive and opportunity to gain access to the device. This threat model is of special importance for smartphone security, since the mobile contexts in which the devices are used are prone to present the most challenging situations in terms of potential exposure to ill-intentioned parties.

But even if less common, attacks by experts are a real threat. These adversaries have the advantage of not having to be co-located with the user, and can therefore target much more people. This is the case with malware. In this other threat model, tap phrase authentication may present new risks.

One of them is the age-old exploitation on key strength, materialized in brute-force, guessing and dictionary attacks. The feasibility of these attacks is mainly dependent on two factors, both of which were not analyzed in this work. The first is the diversity of possible keys, which affects how many guesses it takes to find the secret (Weir et al. 2010). Tap phrases have a theoretically unbounded key space, but since recognition is not exact but based on similarity, information entropy is certainly reduced. The second factor is human-centered. Passwords are made easier to guess because people tend to choose secrets that are easy to memorize (Adams & Sasse 1999), thus reducing the *de facto* key space. A similar effect may be present in tap phrases, whereby people may be choosing tap phrases that map famous songs, jingles or chants.

The proposed technique may also be susceptible to other types of attack, namely capturing what are sometimes called “compromising emanations” (Aviv et al. 2012; Cai & Chen 2011; Foo Kune & Kim 2010; Miluzzo et al. 2012). When a user taps on the

screen, it may be possible to detect the pattern through the device's own microphone, camera or accelerometer. There is also the possibility of external automated observation, for instance through video cameras.

7.3 Future Work

Clear avenues for further research were opened. Future worked is planned in several threads:

- Extending the tap phrase recognizer to be a more general tap phrase library. A clear opportunity for expanding the utility of the work in Chapter 3 is creating a tap phrase dictionary component that can be easily plugged in other applications. Some pieces created in the process of developing the proof-of-concept application already pack much of the functionality needed for configuring tap phrases by demonstration. The recognizer, however, needs to be retooled to instead of making a final decision on matching, providing an n-list of possible matches.
- Analyzing learning effects and skill improvement. During this work, we observed that with prolonged use, people tend to be more accurate in reproducing tap phrases. Wobbrock (2009) suggests that there are “subtle but reliable individual differences in people's tapping” that can be leveraged to prevent others from being able to repeat our tap phrases accurately even when they can observe them. Although we didn't design our studies to test this effect, we suspect that this will only be the case when the users themselves are very accurate. What we observed, in the short experiments we conducted, was that users didn't seem to behave this way. Do users become significantly more accurate with training? This is clearly an empirical question that can be addressed. If the answer is yes, there is an opportunity to personalize tap phrase authentication.
- Characterizing tap phrase choice and associated strength. Although the theoretical security of the tap phrases is favorable, since it has an infinite key space, what ultimately determines the strength of any type of secret is the human element. Further studies, with a broader temporal horizon, are necessary to characterize what types of tap phrases people choose, and why they choose them. It is reasonable to assume that people will try to cope with limitations in memory by using patterns they are familiar with. One dimension that will be explored is comparing the actual variety of tap phrases with that of passwords. This research approach is made possible by the rather large sets of leaked passwords that have come to public.

Bibliography

Adams, A. & Sasse, M.A., 1999. Users are not the enemy. *Communications of the ACM*, 42(12), pp.40–46.

Aviv, A.J. et al., 2012. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference on - ACSAC '12*. New York, New York, USA: ACM Press, p. 41.

Aviv, A.J. et al., 2010. Smudge attacks on smartphone touch screens. In *WOOT'10 Proceedings of the 4th USENIX conference on Offensive technologies*. Berkeley, CA: USENIX Association, pp. 1–7.

Azenkot, S. et al., 2012. PassChords: secure multi-touch authentication for blind people. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '12*. New York, New York, USA: ACM Press, p. 159.

Balzarotti, D., Cova, M. & Vigna, G., 2008. ClearShot: Eavesdropping on Keyboard Input from Video. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, pp. 170–183.

Bargas-Avila, J.A. & Hornbæk, K., 2011. Old wine in new bottles or novel challenges: : a critical analysis of empirical studies of user experience. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, p. 2689.

Barkhuus, L., 2012. The mismeasurement of privacy: using contextual integrity to reconsider privacy in HCI. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, p. 367.

Bell, G. & Dourish, P., 2006. Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. *Personal and Ubiquitous Computing*, 11(2), pp.133–143.

Ben-Asher, N. et al., 2011. 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*. New York, New York, USA: ACM Press, p. 465.

Bergadano, F., Gunetti, D. & Picardi, C., 2002. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*, 5(4), pp.367–397.

Bianchi, A., Oakley, I., Kostakos, V., et al., 2011. The phone lock: audio and haptic shoulder-surfing resistant PIN entry methods for mobile devices. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction - TEI '11*. New York, New York, USA: ACM Press, p. 197.

Bianchi, A. & Oakley, I., 2012. Open Sesame: Design Guidelines for Invisible Passwords. *Computer*, 45(4), pp.58–65.

Bianchi, A., Oakley, I. & Kwon, D.S., 2011. Spinlock: a single-cue haptic and audio PIN input technique for authentication. , pp.81–90.

Biddle, R., Chiasson, S. & Van Oorschot, P.C., 2012. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys*, 44(4), pp.1–41.

Böhme, R. & Grossklags, J., 2011. The security cost of cheap user interaction. In *Proceedings of the 2011 workshop on New security paradigms workshop - NSPW '11*. New York, New York, USA: ACM Press, p. 67.

Bonneau, J. & Preibusch, S., 2010. The password thicket: technical and market failures in human authentication on the web.

Braz, C., 2011. *Integrating a usable security protocol for user authentication into the requirements and design process*. Université du Québec à Montréal.

Brumby, D. & Seyedi, V., 2012. An empirical investigation into how users adapt to mobile phone auto-locks in a multitask setting. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12*. New York, New York, USA: ACM Press, p. 281.

Bulling, A., Alt, F. & Schmidt, A., 2012. Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, p. 3011.

Cai, L. & Chen, H., 2011. TouchLogger: inferring keystrokes on touch screen from smartphone motion. In *Proceedings of the 6th USENIX conference on Hot topics in security (HotSec'11)*. Berkeley, CA: USENIX Association, p. 9.

Chiasson, S., Biddle, R. & van Oorschot, P.C., 2007. A second look at the usability of click-based graphical passwords. In *Proceedings of the 3rd symposium on Usable privacy and security - SOUPS '07*. New York, New York, USA: ACM Press, p. 1.

Chin, E. et al., 2012. Measuring user confidence in smartphone security and privacy. In *Proceedings of the Eighth Symposium on Usable Privacy and Security - SOUPS '12*. New York, New York, USA: ACM Press, p. 1.

Church, K. & Oliver, N., 2011. Understanding mobile web and mobile search use in today's dynamic mobile landscape. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11*. New York, New York, USA: ACM Press, p. 67.

Clarke, N.L. & Furnell, S.M., 2005. Authentication of users on mobile telephones – A survey of attitudes and practices. *Computers & Security*, 24(7), pp.519–527.

Connelly, K., Khalil, A. & Liu, Y., 2007. Do i do what i say?: observed versus stated privacy preferences. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction (INTERACT'07)*. Berlin, Heidelberg: Springer-Verlag, pp. 620–623.

Cranor, L.F. & Garfinkel, S., 2005. *Security and Usability: Designing Secure Systems that People Can Use*, O'Reilly Media, Inc.

Dhamija, R., Tygar, J.D. & Hearst, M., 2006. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*. New York, New York, USA: ACM Press, p. 581.

Dourish, P. & Anderson, K., 2006. Collective Information Practice: Exploring Privacy and Security as Social and Cultural Phenomena. *Human-Computer Interaction*, 21(3), pp.319–342.

Dunphy, P., 2013. *Usable, Secure and Deployable Graphical Passwords*. Newcastle University.

Dunphy, P., Heiner, A.P. & Asokan, N., 2010. A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security - SOUPS '10*. New York, New York, USA: ACM Press, p. 1.

Falaki, H. et al., 2010. Diversity in smartphone usage. In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*. New York, New York, USA: ACM Press, p. 179.

Felt, A.P., Egelman, S. & Wagner, D., 2012. I've got 99 problems, but vibration ain't one: A Survey of Smartphone Users' Concerns. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '12*. New York, New York, USA: ACM Press, p. 33.

Findling, R.D. & Mayrhofer, R., 2012. Towards face unlock: on the difficulty of reliably detecting faces on mobile phones. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia - MoMM '12*. New York, New York, USA: ACM Press, p. 275.

Florencio, D. & Herley, C., 2007. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*. New York, New York, USA: ACM Press, p. 657.

Foo Kune, D. & Kim, Y., 2010. Timing attacks on PIN input devices. In *Proceedings of the 17th ACM conference on Computer and communications security - CCS '10*. New York, New York, USA: ACM Press, p. 678.

Forget, A., Chiasson, S. & Biddle, R., 2010. Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, p. 1107.

Garfinkel, S., 2005. *Design principles and patterns for computer systems that are simultaneously secure and usable*. Massachusetts Institute of Technology.

Ghomi, E. et al., 2012. Using rhythmic patterns as an input method. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, pp. 1253–1253–1262–1262.

Golle, P., 2006. Revisiting the uniqueness of simple demographics in the US population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society - WPES '06*. New York, New York, USA: ACM Press, p. 77.

Griaule Biometrics, 2009. *Understanding Biometrics (Ebook)*,

Grudin, J., 1994. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM*, 37(1), pp.92–105.

Grudin, J. & Poltrock, S., 2012. CSCW - Computer Supported Cooperative Work. In M. Soegaard & R. F. Dam, eds. *Encyclopedia of Human-Computer Interaction*. Aarhus, Denmark: The Interaction Design Foundation.

Guerreiro, T. et al., 2008. From Tapping to Touching: Making Touch Screens Accessible to Blind Users. *IEEE Multimedia*, 15(4), pp.48–50.

Hassenzahl, M., 2004. The Interplay of Beauty, Goodness, and Usability in Interactive Products. *Human-Computer Interaction*, 19(4), pp.319–349.

Herley, C., 2009. So long, and no thanks for the externalities: The Rational Rejection of Security Advice by Users. In *Proceedings of the 2009 workshop on New security paradigms workshop - NSPW '09*. New York, New York, USA: ACM Press, p. 133.

Hwang, S., Cho, S. & Park, S., 2009. Keystroke dynamics-based authentication for mobile devices. *Computers & Security*, 28(1-2), pp.85–93.

Inglesant, P.G. & Sasse, M.A., 2010. The true cost of unusable password policies. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, p. 383.

Jermyn, I. et al., 1999. The design and analysis of graphical passwords. In *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8 (SSYM'99)*. USENIX Association, p. 1.

Khot, R.A., Kumaraguru, P. & Srinathan, K., 2012. WYSWYE: shoulder surfing defense for recognition based graphical passwords. In *Proceedings of the 24th Australian Computer-Human Interaction Conference on - OzCHI '12*. New York, New York, USA: ACM Press, pp. 285–294.

Klasnja, P. et al., 2009. "When I am on Wi-Fi, I am fearless": privacy concerns & practices in eeryday Wi-Fi use. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. New York, New York, USA: ACM Press, p. 1993.

Komanduri, S. et al., 2011. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, p. 2595.

Kumar, M. et al., 2007. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security - SOUPS '07*. New York, New York, USA: ACM Press, p. 13.

Kwasny, M. et al., 2008. Privacy and Technology: Folk Definitions and Perspectives. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08*. New York, New York, USA: ACM Press, p. 3291.

Lee, S. & Zhai, S., 2009. The performance of touch screen soft buttons. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. New York, New York, USA: ACM Press, p. 309.

Lehikoinen, J.T., Lehikoinen, J. & Huuskonen, P., 2007. Understanding privacy regulation in ubicomp interactions. *Personal and Ubiquitous Computing*, 12(8), pp.543–553.

Li, Y., 2010. Protractor: a fast and accurate gesture recognizer. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, p. 2169.

Lin, F.X., Ashbrook, D. & White, S., 2011. RhythmLink: securely pairing I/O-constrained devices by tapping. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. New York, New York, USA: ACM Press, p. 263.

De Luca, A., 2011. *Designing Usable and Secure Authentication Mechanisms for Public Spaces*. lmu.

De Luca, A. et al., 2012. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, pp. 987–987–996–996.

De Luca, A., Denzel, M. & Hussmann, H., 2009. Look into my eyes!: can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security - SOUPS '09*. New York, New York, USA: ACM Press, p. 1.

De Luca, A., Weiss, R. & Drewes, H., 2007. Evaluation of eye-gaze interaction methods for security enhanced PIN-entry. In *Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments - OZCHI '07*. New York, New York, USA: ACM Press, p. 199.

De Luca, A., Weiss, R. & Hussmann, H., 2007. PassShape – Stroke based Shape Passwords. In *Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments - OZCHI '07*. New York, New York, USA: ACM Press, p. 239.

Maggi, F. et al., 2011. Poster:fast, automatic iPhone shoulder surfing. In *Proceedings of the 18th ACM conference on Computer and communications security - CCS '11*. New York, New York, USA: ACM Press, p. 805.

Maguire, J. & Renaud, K., 2012. You only live twice or the years we wasted caring about shoulder-surfing. In *Proceedings of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers (BCS-HCI '12)*. Swinton, UK: British Computer Society, pp. 404–409.

Marques, D. et al., 2013. Under the Table: Tap Authentication for Smartphones. In *Proceedings of BCS HCI – The Internet of Things XXVII*. Uxbridge, UK: British Computer Society.

Marques, D., Duarte, L. & Carriço, L., 2012. Privacy and secrecy in ubiquitous text messaging. In *Proceedings of the 14th international conference on*

Human-computer interaction with mobile devices and services companion - MobileHCI '12. New York, New York, USA: ACM Press, p. 95.

Miluzzo, E. et al., 2012. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12*. New York, New York, USA: ACM Press, p. 323.

Muslukhov, I. et al., 2013. 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*. New York, New York, USA: ACM Press, p. 271.

Muslukhov, I. et al., 2012. Understanding Users' Requirements for Data Protection in Smartphones. *2012 IEEE 28th International Conference on Data Engineering Workshops*, pp.228–235.

Oulasvirta, A. et al., 2005. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*. New York, New York, USA: ACM Press, p. 919.

Payne, B.D. & Edwards, W.K., 2008. A Brief Introduction to Usable Security. *IEEE Internet Computing*, 12(3), pp.13–21.

Perković, T. et al., 2011. Breaking undercover: exploiting design flaws and nonuniform human behavior. In *Proceedings of the Seventh Symposium on Usable Privacy and Security - SOUPS '11*. New York, New York, USA: ACM Press, p. 1.

Porter, S.N., 1982. A password extension for improved human factors. *Computers & Security*, 1(1), pp.54–56.

Saltzer, J.H. & Schroeder, M.D., 1975. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9), pp.1278–1308.

Sauro, J. & Dumas, J.S., 2009. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. New York, New York, USA: ACM Press, p. 1599.

Van Schaik, P., Hassenzahl, M. & Ling, J., 2012. User-Experience from an Inference Perspective. *ACM Transactions on Computer-Human Interaction*, 19(2), pp.1–25.

Schaub, F., Deyhle, R. & Weber, M., 2012. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia - MUM '12*. New York, New York, USA: ACM Press, p. 1.

Shinohara, K., 2010. Investigating meaning in uses of assistive devices: implications of social and professional contexts. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '10*. New York, New York, USA: ACM Press, p. 319.

Shinohara, K. & Wobbrock, J.O., 2011. In the shadow of misperception: assistive technology use and social interactions. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, p. 705.

Shye, A., Scholbrock, B. & Memik, G., 2009. Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture - Micro-42*. New York, New York, USA: ACM Press, p. 168.

Sieger, H. & Möller, S., 2012. Gender differences in the perception of security of mobile phones. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion - MobileHCI '12*. New York, New York, USA: ACM Press, p. 107.

Spiekermann, S., Grossklags, J. & Berendt, B., 2001. E-privacy in 2nd generation E-commerce: privacy preferences versus actual behavior. In *Proceedings of the 3rd ACM conference on Electronic Commerce - EC '01*. New York, New York, USA: ACM Press, pp. 38–47.

Tari, F., Ozok, A.A. & Holden, S.H., 2006. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of the second symposium on Usable privacy and security - SOUPS '06*. New York, New York, USA: ACM Press, p. 56.

Tronci, R. et al., 2011. Fusion of multiple clues for photo-attack detection in face recognition systems. In *2011 International Joint Conference on Biometrics (IJCB)*. IEEE, pp. 1–6.

Uesugi, S., Okada, H. & Sasaki, T., 2010. The impact of personality on acceptance of privacy-sensitive technologies: A comparative study of RFID and finger vein authentication systems. In *2010 IEEE International Symposium on Technology and Society*. IEEE, pp. 111–122.

Weir, M. et al., 2010. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proceedings of the 17th ACM conference on Computer and communications security - CCS '10*. New York, New York, USA: ACM Press, p. 162.

Whitten, A. & Tygar, J.D., 1999. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. , p.14.

Wiedenbeck, S. et al., 2005. PassPoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, 63(1-2), pp.102–127.

Wobbrock, J.O., 2009. TapSongs: tapping rhythm-based passwords on a single binary sensor. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*. New York, New York, USA: ACM Press, p. 93.

Wood, H.M., 1977. The use of passwords for controlling access to remote computer systems and services. In *Proceedings of the June 13-16, 1977, national computer conference on - AFIPS '77*. New York, New York, USA: ACM Press, p. 27.

Zakaria, N.H. et al., 2011. Shoulder surfing defence for recall-based graphical passwords. In *Proceedings of the Seventh Symposium on Usable Privacy and Security - SOUPS '11*. New York, New York, USA: ACM Press, p. 1.

Von Zezschwitz, E. et al., 2013. Making graphic-based authentication secure against smudge attacks. In *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*. New York, New York, USA: ACM Press, p. 277.

Zezschwitz, E. von, Dunphy, P. & Luca, A. De, 2013. Patterns in the Wild: A Field Study of the Usability of Pattern and PIN-based Authentication on Smartphones. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12*. New York, New York, USA: ACM Press.

Zhang, Y. et al., 2012. Fingerprint attack against touch-enabled devices. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '12*. New York, New York, USA: ACM Press, p. 57.

Zhu, F., Carpenter, S. & Kulkarni, A., 2012. Understanding identity exposure in pervasive computing environments. *Pervasive and Mobile Computing*, 8(5), pp.777–794.