A Framework for Continuous, Transparent Authentication on Mobile Devices

Heather Anne Crawford

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Computing Science

College of Science and Engineering
University of Glasgow

December 2012

Abstract

Mobile devices have consistently advanced in terms of processing power, amount of memory and functionality. With these advances, the ability to store potentially private or sensitive information on them has increased. Traditional methods for securing mobile devices, passwords and PINs, are inadequate given their weaknesses and the bursty use patterns that characterize mobile devices. Passwords and PINs are often shared or weak secrets to ameliorate the memory load on device owners. Furthermore, they represent point-of-entry security, which provides access control but not authentication. Alternatives to these traditional methods have been suggested. Examples include graphical passwords, biometrics and sketched passwords, among others. These alternatives all have their place in an authentication toolbox, as do passwords and PINs, but do not respect the unique needs of the mobile device environment.

This dissertation presents a continuous, transparent authentication method for mobile devices called the Transparent Authentication Framework. The Framework uses *behavioral biometrics*, which are patterns in how people perform actions, to verify the identity of the mobile device owner. It is *transparent* in that the biometrics are gathered in the background while the device is used normally, and is *continuous* in that verification takes place regularly. The Framework requires little effort from the device owner, goes beyond access control to provide authentication, and is acceptable and trustworthy to device owners, all while respecting the memory and processor limitations of the mobile device environment.

Acknowledgements

First, to my supervisors, Dr Karen Renaud and Dr Tim Storer. I couldn't have asked for more dedicated people to help me along this path. Thank you for the wisdom, guidance and sympathetic ear. Having two supervisors was sometimes a challenge, but I wouldn't have had it any other way. Thank you to my SICSA supervisor, Dr Mark Dunlop of the University of Strathclyde; mobile devices and user studies are demystified largely due to your help.

No one does this kind of work on their own. I have a lot of people to thank for providing help on various details of the research contained herein. In no particular order, thank yous go to Julie Williamson, Simon Rogers, John Williamson, Marilyn McGee-Lennon, David Masters, and Alessandro Vinciarelli. Special thanks go to John Aycock. You listened, read drafts and provided support even though you're not my supervisor this time!

No small mention goes to the people who participated in the user studies in this work. Without you, this would have been impossible, so thank you. Thank you to SICSA for the studentship and the chance to expand my knowledge through masterclasses, workshops and talks from Distinguished Visitors. I'd also like to thank the University of Glasgow College of Science and Engineering for the scholarship.

Last, but never, ever least: thank you to my husband, Paul. You've stood by me through it all; I can't possibly express how grateful I am for your unwavering love and support.

For Paul.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In 1965, Gordon Moore predicted that the number of transistors on integrated circuits would double every two years [1]. His prediction, now known as *Moore's Law*, has been stated in a more colloquial manner: the processor speed of computers will double every two years. Since its inception, this prediction has guided the computer industry, in terms of both research and manufacturing.

Computers continue to improve. Performance increases in terms of memory, processor speed and functionality with great regularity. This is especially evident with mobile devices. Once simple telephony tools, mobile devices have become fully-fledged computing environments. Their features, functionality and near-constant connection to the Internet and mobile service providers has unbound people from their desktop and laptop computers. This freedom does not, however, come without cost. The improvements in processor speed, amount of memory, functionality and features allow us to work (and play) more than ever before. Accordingly, mobile device popularity has soared – in 2011, 488 million smartphones were sold, which is more than desktop and laptop computers combined [2]. Their ubiquity and features mean mobile devices now store more information than ever before, some of it personal or personally identifying [3]. Furthermore, we have come to depend on them to provide access to services such as email and the Internet, among others, and are often at a loss if they are not present. Due to the nature and amount of data now stored on these devices, a security method for protecting access to this information is required.

1.1 The Problem

The motivation for this work comes from several areas. Modern mobile devices are now able to perform potentially risky tasks such as the ability to store (corporate and personal) data, and e-transactions such as making purchases or online banking. With such broad access to

services comes the ability (and responsibility) to store and access increasingly personal (and personally identifying) information about the device owner and their activities. This in turn indicates the need for a way of protecting this data from those who should not have access to it – the authentication problem.

Current authentication methods are known to have issues with strength and memorability [4, 5]. The real issue is not that passwords are broken – they have their place in a toolbox of authentication schemes, and are particularly useful in situations where humans are excluded. For instance, computers authenticate to each other, and can remember long, complicated passwords with ease. The overarching problem is with humans – the memory-load put onto users to remember several long passwords encourages the use of coping mechanisms such as reuse and sharing [4, 6]. Furthermore, the bursty nature that characterizes mobile device use [7, 8] means that the device owner must enter their password frequently. This represents a significant inconvenience and may encourage the device owner to subvert the security mechanism.

The problems with current authentication methods are informed by the following standing issues in computer security, which also provide a basis for this research:

The Password Problem has been described as the willingness of users, despite advice and requirements to the contrary, to choose weak passwords and share, reuse and write them down. This problem is based on the proposition that "strong" passwords (i.e., those that are difficult to break: long, with various cases, special characters, and numbers) are often difficult to create and to remember. This problem is exacerbated when users require different strong passwords for each of the approximately 25 separate accounts the average user has [9]. Furthermore, passwords and PINs provide binary access control. Once the secret knowledge is entered, access to all protected data and functionality is allowed. In this way, resources are either protected or unprotected; there is no nuanced control over the *level* of protection.

Disconnect between Mental Models and Password Security: This problem refers to the idea that users have a skewed vision of the dangers associated with security methods, especially with password reuse and sharing. Many users do not believe they are at risk, or that they have "anything worth having" [10]. Furthermore, the threats linked with using weak password practices such as identity theft, fraud and account abuse are considered distant threats by many users. There is no conclusive proof that a strong password will protect users from such threats, or, conversely, that a weak password does indeed make them more vulnerable since there is no way to link the possibility of a threat to an actual instance of the threat's occurrence. This disconnect between the mind-model and the security of a password ensures that passwords and similar authentication methods will be adjusted by users to make them more usable.

Inflexibility in Authentication Policy creation: In direct response to fears regarding threats due to password weakness, many organizations have imposed significant authentication policies on their employees. Such policies, in terms of passwords, define the required length, character set and change frequency for passwords used to gain access to company resources. Such policies are known to not only force users to circumvent them in order to cope [4, 11], but also to provide a reduced search space to potential attackers.

Other security methods for mobile devices have been proposed, including sketched passwords[1], biometrics [12–14] and graphical passwords [15, 16]. Each of these solutions also has issues similar to those with passwords and PINs. They are effortful, have memorability issues, and provide point-of-entry protection. The solution to this problem should take these issues into account when proposing an alternative to traditional authentication mechanisms.

1.2 The Solution: Teaching Computers to "Know" Their Owner

Mobile devices are fully-fledged computing platforms, and this has opened up an attack vector that is not effectively managed by current authentication mechanisms. Device owners cope with the current mechanisms in insecure ways. A solution to the mobile device authentication problem is something that is as effortless as possible, and provides protection that goes beyond point-of-entry security. A solution to the mobile device authentication problem should have the following attributes:

1. Require less effort than current authentication methods [17];
2. Go beyond access control and point-of-entry solutions to protect data and functionality at a more granular level [17];
3. Authenticate users continuously to maintain confidence in their identity [17];
4. Provide a security method that is acceptable and considered trustworthy by device owners;
5. Respect the needs of the mobile device environment in terms of its bursty nature as well as its limitations in both processor speed and memory.

[1] `http://support.google.com/android/bin/answer.py?hl=en&answer=2381897`

1.3 The Transparent Authentication Framework

This dissertation introduces the Transparent Authentication Framework: a framework to support the creation of a mechanism that provides continuous, transparent authentication on mobile devices. The Framework uses patterns in how users perform regular device actions to affect the mobile device's knowledge of who is currently using it. In this research, the mobile device's level of certainty that the current user is the device owner is called *device confidence*. Behavioral biometrics, which are patterns in user actions, are used to inform device confidence levels. A biometric match increases device confidence, and a non-match lowers it. The tasks and data on the device are mapped to particular device confidence levels. For instance, highly private data such as a list of passwords may be assigned a high security level. If this level is higher than the current device confidence, then access to the data or functionality is denied. In the event that the device confidence is too low to accomplish a particular task, the legitimate device owner may use an explicit authentication method to increase their device confidence. If the device is no longer being used, the device confidence will lower over time. Eventually, the user will have access to only very basic functionality, but device confidence can be increased again via biometrics or explicit authentication.

The Framework provides a solution to the mobile device authentication problem by addressing each of the attributes given in the previous section. It does so in the following ways:

Reduces user effort by using behavioral biometrics, which can be gathered while the device owner uses the device in their normal manner. Two biometrics were tested for this purpose: keystroke dynamics and speaker verification. The former uses patterns in the way we type and the latter uses patterns in the way we speak;

Goes beyond access control by using biometrics in combination with explicit methods to verify the identity of the device owner;

Authenticates continuously by collecting biometrics and using them regularly to increase device confidence. Storing the biometrics and replacing them with newer samples frequently further supports the continuous nature by allowing recalculation even when the device owner is not currently using the device.

Provides an acceptable and trustworthy security method as evidenced by user studies conducted as part of this research.

Respects the limitations of mobile devices by requiring only the hardware already on the device and minimizing processor and memory use by selecting biometrics and classifiers that are simple and have minimal processing needs.

These contributions are summarized in Table 1.1. The first column identifies the requirements for a transparent authentication method for mobile devices; the second column specifies how the Transparent Authentication Framework meets the requirement. The third column lists an attribute provided by the Framework that meets the requirement in question. The final column shows which chapter of this dissertation contains the explanation or experimental work that supports each stated feature.

Requirement	How Met	Attribute	Chapter
Less user effort	Behavioral biometrics	Transparency	4, 5
Beyond access control	Authentication provision, task mapping	Authentication	3
Continuous authentication	Continuous device confidence recalculation, task mapping	Continuousness	3
Acceptable, trustworthy method	User study into perceptions	Acceptability, trustworthiness	7
Respects mobile environment	Uses minimal hardware and efficient algorithms	Minimality	6, 7

Table 1.1: How the Transparent Authentication Framework meets the needs for a mobile device authentication solution.

The Transparent Authentication Framework is a potential solution to the mobile device authentication problem. Its creation is driven by the research question and hypotheses stated in the next section.

1.4 Research Question

This research is based on the following research question:

> It is possible to verify the identity of the current user of a mobile device in a secure, continuous, transparent and passive manner by using a combination of behavioral biometrics. Such authentication will not normally require explicit owner action, but will instead rely on the owner's usual interaction with the mobile device. Finally, such a transparent authentication method will be acceptable to device owners.

The following assumptions have been made in carrying out this research:

1. Mobile devices are single user devices (this may not be the case in all countries). This assumption reduces the complexity of the overarching problem of owner identification

versus verification. The mobile device user is implicitly claiming a particular identity, that of device owner, when using the device. Therefore, the only biometric patterns that the gathered patterns must be compared to are those of the device owner.

2. Behavioral biometrics are not unique to a specific user. Instead, they are relatively distinctive and stable enough to support authentication in a small population [18–20], especially when combined into multimodal biometrics.

1.4.1 Research Hypotheses

The following research hypotheses are based on the above research question:

H1: Behavioral biometrics such as keystroke dynamics and speaker verification are sufficiently distinctive to contribute to verification of the identity of a mobile device owner.

H2: Combining keystroke dynamics and speaker verification into a multimodal behavioral biometric reduces the error rates seen with the individual biometrics.

H3: It is possible to gather keystroke dynamics and speaker verification biometrics while the mobile device user goes about other tasks on the device.

H4: Mobile device owners would consider using a transparent authentication method if it was available to them.

A framework that combines the above hypotheses is the major contribution this dissertation provides. The assertion in this research is that the Framework is device and operating system independent.

1.5 Main Contributions of this Research

This research contributes new knowledge to the field of mobile device security. Specifically, it provides the design for a framework upon which continuous, transparent mobile device security may be based. The Transparent Authentication Framework goes beyond other similar models by keeping the owner's private, identifying information on the device and making all decisions regarding identity on–device. Furthermore, the Framework uses multimodal biometrics to overcome some of the limitations of single biometrics, and allows the developer who uses the Framework to choose not only the type but also the number of biometrics to include. Finally, this Framework allows the *user* to control the mapping of security level to the tasks and data available on the device; in other similar work, this is left to the developer.

The following publications have resulted from exploring the research areas described in this dissertation, as follows:

Heather Crawford and Karen Renaud, "Invisible, Passive, Continuous, and Multimodal Authentication". In *Proceedings of the Mobile Social Signal Processing Workshop*, 2010, to appear.

Heather Crawford, "Keystroke Dynamics: Characteristics and Opportunities". In *Proceedings of the 8th Annual Conference on Privacy, Security, and Trust (PST)*, 2010, pp. 205 – 212.

The following papers related to this research are currently under peer review:

Heather Crawford, Karen Renaud and Tim Storer. "A Framework for Continuous, Transparent Mobile Device Authentication". Submitted to *Computers & Security* Special Issue on Active Authentication. (Under revision).

1.6 Dissertation Structure

This dissertation continues with a discussion of the background needed to understand the studies and research performed for this work, including an overview of the state-of-the-art in authentication research. Next, the Transparent Authentication Framework is presented in detail in Chapter 3. Then, the four user studies undertaken to justify the Framework's inclusions are presented. These four feasibility studies examine keystroke dynamics (Chapter 4), speaker verification (Chapter 5), combining biometrics into multimodal authenticators (Chapter 6), and finally a study to gather user perceptions of transparent authentication (Chapter 7). Finally, the security issues inherent in the Framework are discussed in Chapter 8, and the conclusions and future work appear in Chapter 9.

Chapter 2

Background

This chapter introduces concepts and current research in the field of authentication. Topics covered begin with a discussion of current authentication methods and a discussion of the issues caused by widespread password use. Then, alternatives to passwords and their acceptance by users are discussed. The focus of this research is on mobile device authentication, so subsequent sections focus on methods used on mobile devices. Biometrics, including physiological, behavioral and multimodal are then discussed, along with user acceptance of them, and methods of measuring biometric performance. The focus of the chapter then shifts to transparent authentication mechanisms and frameworks that support them, which often use biometrics as a basis. Finally, pattern classification concepts and research are discussed since they can be used to support biometric decision-making. The chapter concludes with a description of the Transparent Authentication Framework and the terminology used throughout the dissertation. This chapter extends the motivation discussion given in the previous chapter.

2.1 Authentication and Access Control

Authentication and access control are linked concepts that are part of information and system security. *Authentication* verifies the identity of one person, process or computer to another. *Access control* determines what a person, process or computer may do with the resources mediated by another person, process or computer. Access control generally requires identification followed by an authentication step that confirms the validity of the claimed identity. It is used as a means of limiting resource access to those who are pre-approved [21], and as a means of implementing a measure of accountability when using the protected resources.

The access control problem has three components: identification, authentication and authorization [22]. This chapter (and research) is concerned with the first two of these components. User authentication, a special case of the broader topic of authentication, begins when a user

claims an identity, either explicitly by providing a username or a card with a chip that holds an identity, or implicitly by possessing a device. Next, the user provides some evidence to support this claim. This evidence is used to authenticate the user; if successful, the user is granted access to a protected resource. The authorization component mediates this stage by determining what resources may be accessed.

Authentication mechanisms are traditionally built upon one or more of the following three types of factors [22]:

1. *Something you know:* This is a secret that the user shares with the authentication system, such as a password, PIN or answer to a challenge question. This factor is known as a *secret-knowledge technique*. This recall-based method is often used as a form of authentication despite the fact that it can allow access to anyone who knows the shared secret rather than to a specific person. Secret knowledge can also be easy to share and to guess.

2. *Something you have:* These are usually tokens such as a smartcard, RFID chip, keyfob or other hardware token. This factor can be combined with something you know to provide additional security. Physical objects such as this can be easy to share with others, and can be lost or stolen. To manage theft and loss, there must be a method of canceling those tokens that are no longer possessed by the intended owner, which adds complexity to systems that use them. Users tend to find tokens cumbersome and inconvenient despite their widespread use [23].

3. *Something you are:* Biometrics, both physiological and behavioral, can be used to support authentication. The latter is also referred to as *something you do*. Biometrics can be more difficult to impersonate or forge compared to knowledge or possessions, but are computationally more difficult to process. They can require more hardware than other methods, although behavioral biometrics often do not. Physiological examples include fingerprints, iris and retina scans and facial recognition. Behavioral examples include typing, voice–related and device use patterns.

These types of factors are related to each other as shown in Figure 2.1. For instance, keystroke dynamics measures typing patterns, and can be combined with secret knowledge entry, such as typing a password. Behavioral biometrics are an example of both something you are and know since our experiences and skills affect how we perform such actions, such as typing.

As research into authentication has progressed, more factors have been suggested, such as the following:

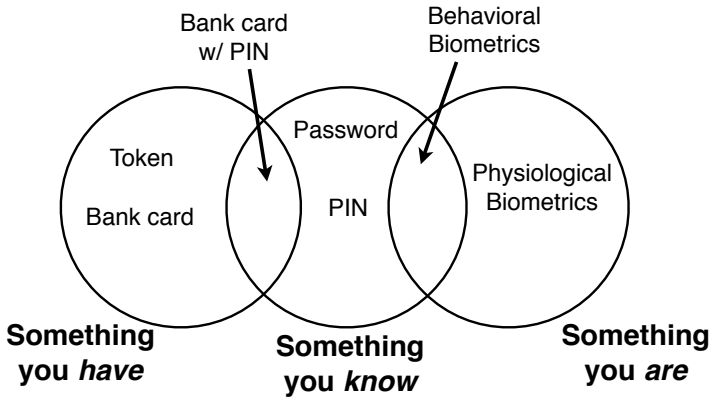


Figure 2.1: Relationship between the three access control factors.

1. *Some**one** you know:* Using Bluetooth or some other short range communication method, the general area around the user is searched for people (or their hardware) who appear in the user's social network [24]. These people are then asked to vouch for the user by confirming their identity.

2. *Where you are:* This factor encompasses location-based activities [25], particularly in ubiquitous computing environments [26]. These can take the form of comparisons to usual locations (i.e., if a person is in a location they visit frequently as opposed to someplace they have never been) or comparison of known calendar events to location.

The factors listed here are often combined into multi-factor authentication solutions to increase the security provided by any one method, and to support their known weaknesses. Passwords and PINs in particular are used almost ubiquitously even though they have significant issues both in design and use that make them a poor choice for security provision in many situations.

2.2 Textual Passwords and the Password Problem

Passwords and other secret knowledge techniques are the most commonly-deployed authentication mechanism despite several problems [22]. They are familiar to users, and may not compromise privacy provided the user does not use publicly-known information in their password choice. However, there is a well-understood trade-off, known as the password problem, between the security of a password (i.e., the difficulty for an attacker in guessing the secret) and the memorability of the password for the user. Typically, the harder a password is to guess for an attacker, the harder it may be for a legitimate user to remember. This trade-off between memorability and security has encouraged research into password strengthening and improved memorability in the form of alternatives to traditional passwords.

Efforts into improving memorability and security have included password phrases [27] and mnemonic passwords [28], both of which attempt to create secure passwords with built–in memorability aids. Other research focuses on balancing rather than improving memorability and security, such as using persuasive technology to encourage users to select secure and memorable passwords [29]. In this approach, users are allowed to select a password and then additional characters are added at random positions to improve the password's security. The users are then allowed to shuffle the characters to find a combination that is memorable. The result was that users chose more secure passwords overall, but they still tended to choose weak initial passwords to improve memorability.

Despite strong research interest in improving passwords and secret-knowledge techniques in general, there has been no single authentication mechanism of this type that is considered both secure and usable. It is likely, then, that rather than focusing on finding the single authentication mechanism that will be the panacea to all authentication needs, research should focus on creating a toolbox of possible authentication methods that can be selected to suit a particular application's needs. To this end, research into password alternatives has become an increasingly important field.

2.3 Alternatives to Passwords

The problems with traditional authentication mechanisms has not gone unnoticed in the research community; many alternatives to traditional textual passwords have been suggested. These include, but are not limited to, the following ideas:

Graphical passwords. This method relies on using either user-chosen or system-selected images to authenticate the user. *Click-based* graphical passwords [15, 30] consist of a series of n points on an image or series of images that the user has chosen during the enrolment process. During enrolment, the user chooses an image, then clicks on a series of memorable points on the image. The password is entered by subsequently clicking on the same points in the same order, within certain tolerances. Other methods require the user to select m prechosen images from a series of $k > m$ distractor images [31, 32], or to draw a simple sketch on a grid of known size [33, 34]. Graphical passwords, while shown via lab studies to be usable and acceptable to users [35, 36], have not gained significant notice or use outside of laboratory studies. The reasons for this are not yet known, although Chiasson *et al.* postulate that it is because laboratory studies do not accurately mimic real-world use [36]. Based on the results of a lab study into the usability of graphical passwords, Stobert *et al.* [37] have argued that graphical passwords are potentially a useful security measure for mobile devices, both in terms of usability and expected security level. Their lab study did not explicitly in-

clude usability studies for mobile devices, but instead opined that the ability to use a smaller image on a touch screen would lend itself well to a mobile environment.

Single Sign-On (SSO). Some SSO systems use a single strong password to secure a list of other passwords in order to reduce the user's memory load. Detractors of password-based SSO systems note that the loss of the main password, no matter how strong it is considered, reveals all passwords it protects, and thus the other passwords are useless. The security level would be the same if the strong password were used on each of the accounts it protects rather than having different passwords. The hope of SSO supporters is that the main password will be strong enough and treated with enough respect that it will not be lost. This is a simple case of moving the security of each individual password to the main password. Since SSO systems can be complex in terms of overhead and initial setup, the lack of additional security over a single strong password is often not seen as worth the overhead SSO provides [38].

Transparent authentication. This method gathers samples of user behavior while going about other tasks on a computer to produce a behavioral use pattern that can be used to verify the identity of the person using the resource. In this scenario, the user does not have to explicitly provide a sample for authentication (other than during enrolment in the authentication system); the expected benefit is that this may reduce frustration and improve security [39]. Transparent authentication may be implemented using biometrics, particularly behavioral since they are often easier to gather implicitly compared to physiological biometrics.

These alternatives to passwords and PINs are positive steps towards finding a viable authentication method for mobile devices. However, research has shown that these methods, as well as passwords and PINs, are not always accepted by users [40, 41].

2.4 User Acceptance of Secret-Knowledge Mechanisms

There has been a considerable amount of research that investigates the extent to which users will opt for secret-knowledge mechanisms, and the extent to which they understand their limitations. In a user study of mobile phone authentication practices, Kowalski and Goldstein found that users did not understand the security options available on mobile devices [42], specifically the difference between (and the existence of) the SIM PIN and the phone security code. Kowalski and Goldstein found that only 32% of users in their study were aware of the SIM PIN, and none of them chose to use it. Similarly, Botha *et al.* [43] distinguish between SIM and handset PINs and recognize that these are simply point-of-entry security mechanisms that have limited ability to provide content security. Botha *et al.* also state that

PIN entry on mobile platforms may be tedious and annoying to the owner because "mobile users may simply wish to take the device out of their pocket to check a schedule entry and could therefore find that entering the password takes longer than the task itself." [43, p. 3]. These concepts suggest the need for a more nuanced and effortless mechanism for mobile devices, as stated explicitly by Botha *et al.*

In a similar study, Clarke and Furnell found that 42% of respondents believed phone security codes (i.e., the handset PIN) provided an "adequate" level of security [44]. Despite the fact that fewer than half of respondents felt the security provided by the PIN was adequate, 66% of respondents used phone security code authentication when first starting up their device, and 18% also used it to awake from standby. These statistics present an impression of users' mental model of security – fewer than half of respondents felt that their device was adequately protected, and yet a significant number regularly "secure" their device with a phone security code. This may be due to users choosing a "something is better than nothing" approach to security, in which they choose to use what is available despite their perceptions of its inadequacy. In a follow-up study, Karatzouni *et al.* [23] confirmed these findings, and state that users did not believe they had anything worth protecting on their mobile device.

While the alternatives to passwords discussed in this section help provide support for a toolbox of authentication mechanisms, they all have issues that prevent them from being the best choice for all authentication needs. In particular, the popularity and ubiquity of smartphones has increased the need for authentication mechanisms that are specifically tailored to these devices. One compelling reason is that they are increasingly able to store and transmit personal information [3], and their mobile nature and susceptibility to loss and theft make them particularly difficult to protect.

2.5 Mobile Device Authentication

The previous section has argued for mobile device authentication methods that are tailored to the nature of the device. One difference between desktop or laptop computers and mobile devices is that they are used very differently. Mobile device use patterns are often characterized by short, bursty intervals [7, 8, 45]. This means that mobile device owners tend to use their device frequently, but for short periods of time. Current mobile device authentication methods, including those discussed below, do not lend themselves well to frequency. Having to enter knowledge-based access control frequently may cause device owners to disable the security mechanism to reduce frustration.

Mobile devices are generally single-user devices, as evidenced by the lack of a multi-user model in the major mobile operating systems. Thus, access control is reduced to verification since the user assumes the identity of the device owner while using it. Many methods of

controlling non-owner access to a mobile device have been studied. Two common methods are passwords and PINs, including the Android sketched password. Quite apart from the known weaknesses of knowledge-based mechanisms, they are of limited utility given that they only protect the device at point-of-entry. This means that once the password or PIN has been verified, the device can be used to its full extent. Furthermore, in this context, since knowledge-based authentication verifies knowledge of a secret and not the identity of the knowledge bearer, this kind of mechanism is somewhat unsatisfactory. If passwords for other applications are stored on the device, then a potential intruder also has access to these applications without authenticating further.

Like other computer systems, passwords and PINs are commonly-used security provisions on mobile devices [44]. There are two types of mobile device PINs [44, 46]; the handset PIN, which protects the handset itself and the data stored in its memory from unauthorized use, and the SIM PIN, which protects the use of and data stored on the SIM card. The handset PIN is the one that most users think of when asked about a mobile device PIN; many people do not realize that in using only this PIN, they are leaving unprotected a significant amount of potentially private information stored on the SIM card. As an example of the difference between the two PINs, note that even with the handset PIN enabled, it is possible to remove a device SIM and use it in another device.

In addition to PINs, which can have a variable number of digits, some mobile devices allow the use of standard alphanumeric passwords. These are different from PINs not only in length, but in possible character sets. Having a larger set of characters to choose from allows for more possible passwords. However, as was discussed previously, passwords on all platforms fall victim to the struggle between memorability and security. This holds on mobile devices as well.

In an attempt to move away from alphanumeric passwords and PINs, some device manufacturers have employed a sketch-based password, in which the user joins a series of points on a grid in a sequence. The order of the points defines the password. While quite memorable, they are also quite insecure since the drawn pattern has limited variations, and can be cracked by looking at the traces left on the screen and through direct observation [47].

Other manufacturers have experimented with graphical passwords, and a few have begun to examine the use of biometrics [48]. However, these experiments are in early stages and currently point towards the need for alternatives to passwords and PINs for protecting mobile devices.

Hardware tokens have been suggested for use in mobile device authentication. With this method, authorized users carry a small, physical object such as a keyfob or card that may be used in combination with a knowledge-based mechanism to authenticate the holder. Tokens may be used to authenticate to other computing systems such as ATM machines, but research

in this area also considers how a token owner can first authenticate to the token, then allow the token to manage further authentication requests from other computing systems. The first part of this dual authentication is of interest in this research; the second is out of scope. Since carrying a token may be seen by users as limiting [23] because it may be forgotten or lost, some researchers have focused on embedding a hardware token in a device that users already carry [49]. Suggestions for tokens have included watches, jewellery, credit card-style cards [50] and mobile devices [51, 52]. Stajano [53] used the mobile device itself as the token, however none of the research to date has investigated this from a transparent perspective. Transparency implies that the device owner does not explicitly provide information to authenticate, but is authenticated via information gathered while they go about other tasks on the device.

Other researched authentication methods include user's social group [24, 54] (e.g., the people nearby who know the user can vouch for their identity) and computer use patterns [55, 56]. This latter area of research takes advantage of the distinctive ways in which people use computers, including mobile devices. For instance, the regular pattern of one person may be to check their email once per hour, and work on a word processing document in between these checks. For another, they may check their email more regularly and use a wider variety of programs in between checks. The research in this field focuses on determining whether these patterns of device use may be sufficient to verify the identity of the device user to a sufficient level of confidence.

2.6 Biometrics

Biometrics is defined as the "science of recognizing an individual based on her physiological and behavioral traits" [57]. A *physiological* biometric is one that is measured from the human body. Examples include iris and retina scans, fingerprints and facial recognition. *Behavioral* biometrics rely on a person's unique behaviors, i.e., how they do particular tasks. Key behavioral biometrics are signatures, gait, voice and keystroke dynamics.

Biometric traits are inextricably linked to the person who provided them since they cannot be shared and are unlikely to be stolen[1], unlike passwords and PINs. Biometrics present an alternative to traditional knowledge- and ownership-based authentication (i.e., something you know and something you have, respectively). Support for biometrics has centered around memorability – biometrics cannot be forgotten. However, they are not universal: for example, approximately 2% of the U.S. population does not have viable fingerprints [58]. Further-

[1]The word "stolen" in terms of a biometric pattern is not the same as "copied". A person's biometric trait may be copied after it is gathered, but in order to steal a physiological biometric, the thief must be in possession of the thing from which the biometric was gathered, for instance a finger or an eye.

more, there are issues with biometric matching ability. Women's fingerprints are harder to match than men's; they require about 150% the processing power of male fingerprints [58].

Biometrics are used for two purposes, as follows [59]:

1. *Verification (a.k.a. authentication):* the person providing the gathered biometric claims an identity, then the gathered biometric is compared to that of the claimed identity in the database. If they match, the person is granted access to the protected resource; otherwise they are not. Verification is called *1:1 matching*. A person may claim an identity by providing a username, or using an identification card either with or without a chip that contains further information. Cards that contain chips, known as *smart-cards*, may also contain the known biometric for the card's owner, which can reduce pattern matching times since the biometric system need not store patterns for all authorized users. For example, a bank may decide to use biometrics in conjunction with chipped debit or credit cards at its ATM machines. When the person inserts the card into the ATM, their fingerprint pattern is accessed from the chip on the card. Then, the ATM prompts the person to scan their fingerprint using the built–in scanner on the ATM. If the gathered and stored patterns match, the person may continue with their transaction. If there is no match, the transaction is not allowed. In this way, the ATM machines need not store a fingerprint for each bank customer in each ATM worldwide, or in a bank server.

2. *Identification:* recognizing a person who has not made a prior claim of identity via a match between the person's offered biometric and any of those in a database of authorized people. The gathered biometric is compared to each pattern in the database until a match is found. If none exists, the person is rejected as unauthorized. Identification is called *1:N matching*. This is a significantly more difficult and time–consuming process when compared to verification, but requires less information from the person offering the biometric. Identification can be used for "negative recognition", which is when the biometric system determines if the person is who they implicitly or explicitly *deny* to be [60]. Logically, negative recognition is the opposite of verification and can be thought of as explicitly denying to be all people in the database except that person whose pattern matches the gathered biometric. Negative recognition is used to prevent people from claiming more than one identity. It can also be used in the case where the subject is not in a database. For instance, fingerprints may be used to identify a person who has been arrested in the past. In this case, a subject who has not been arrested previously claims that their identity is *not* in the fingerprint database.

Like traditional authentication mechanisms, biometrics also have issues that prevent them from being the method of choice in some cases. Jain *et al.* have proposed a list of desirable characteristics for biometrics that help address some of these issues, as follows [14, 59]:

Universality: the majority of system users should have the characteristic being used as a biometric.

Distinctiveness: the characteristic should differ sufficiently between users to allow for identity verification.

Permanence: the characteristic should not vary significantly over time.

Collectability: characteristics must be measured quantitatively.

Performance: this encompasses concerns with accuracy, speed of matching, resources required and operational factors. These should fit the needs of the system under design.

Acceptability: refers to how willing people are to accept the biometric in their daily lives. This trait also encompasses whether they consider the biometric a trustworthy authentication method.

Circumvention: indicates how easily a particular biometric can be used to fraudulently grant access to someone other than the biometric's owner.

Most biometrics have several issues that make them poor choices for some users. A lack of universality leads to *failure to enrol* issues, where certain users may not use a biometric system because they cannot provide biometrics for comparison purposes. *Failure to capture* issues arise when performance degrades or has not been considered fully during the design process. Research into compensating for the variability and issues with individual biometrics has led to efforts into combining them into *multimodal* biometrics. Such combinations can often eliminate the weaknesses of individual biometrics.

Biometrics, both physiological and behavioral, have a strong history as an authentication and access control tool. The following sections introduce current work on the use of biometrics as an authentication method specifically for use with mobile devices. It does not include a discussion of biometrics on desktop or laptop computers. Thorough treatments of these topics can be found in [57, 59, 61–63].

2.6.1 Physiological Biometrics

Examples of physiological biometrics include fingerprints, palmprints, iris and retina scans and facial recognition. These biometrics generally require a method of obtaining them such as a scanner or camera, and a method of converting them from a detailed scan to a concise feature vector that represents the most salient (i.e., distinctive) parts. Each of these biometrics has strengths and weaknesses and the selection of which to use depends on the needs of the application that will deploy it. The biometrics listed above are discussed briefly below.

Fingerprint: the pattern of ridges and valleys on the fingertips. Fingerprints are distinct from finger to finger on the same person, and are different for identical twins. The accuracy of fingerprints in terms of verification is high – around 90% for a single index fingerprint [58]. Fingerprint readers are increasingly affordable (around USD $20 each for large orders) and are appearing on electronics such as laptops, tablet computers and mobile phones [48]. A detractor for fingerprint use, however, is that processing and matching the print requires large amounts of resources such as processor speed and memory, particularly when used for identification versus verification. Another potential issue is permanence, since fingerprints are sensitive to age, damage to the fingertip or loss of the finger. Once these are lost, the fingerprint cannot be used for identification. Furthermore, if a copy of the scanned fingerprint is stolen, it cannot be reset as can a password.

Palmprint: Much like fingerprints, the palm of the hand has ridges and valleys that are distinctive from person to person. The larger surface area of the palm compared to the fingertip is expected to provide more distinctive measurements. The tradeoff is an increase in pattern complexity and thus in matching. The larger palm area means that the scanners must be larger and likely more expensive. They are known to be distinctive [64], particularly when a high-resolution scanner is used. Palmprints are vulnerable to loss or injury to the hand, and are irreplaceable, much like fingerprints. Palmprint biometrics may be combined with scans of the veins in the hand to increase distinctiveness.

Retina: the blood vessels in the back of the eye have a distinctive pattern from person to person, including identical twins [63]. These patterns remain unchanged over a person's life, unlike fingerprints. Retina biometrics are highly accurate and matching is quick. The subject must stand very close to a expensive and specialized scanner and remove glasses or contact lenses, which may limit acceptability. Retinal patterns may be altered by injury or medical conditions such as glaucoma and diabetes. Furthermore, they are affected by severe astigmatism and cataracts, which may render them indistinctive in the elderly.

Iris: the colored region of the eye that is bounded by the pupil and the white area. It is a distinctive biometric pattern (even between twins) that can be gathered from a short distance using a dedicated scanner [63]. The iris can be scanned through glasses and contact lenses, although accuracy may be improved if they are removed. Commercially-deployed iris recognition systems are fast and accurate and are becoming more common-place. It is difficult to surgically alter the iris, and copies are quickly detected. However, irises can be damaged or lost due to injury or loss of the eye itself and are not easily replaced.

Face: is a common recognition method that people use regularly to identify those they have met in the past. The electronic version of facial recognition uses location and shape of facial features such as eyes, nose, eyebrows, lips and chin to create a distinctive pattern. The subject's orientation to the camera, facial expression and lighting in the gathered images are known to be issues with facial recognition. Accuracy is known to be reasonable [64]. Facial recognition may be susceptible to circumvention via photographs held up to the camera.

Table 2.1 shows how each of these biometrics meets Jain's seven biometric characteristics. For each biometric, a determination for its adherence to the characteristic in question has been made. The determinations are High (H), Medium (M) and Low (L), and have been selected by the author. For example, a determination of H for fingerprint universality implies that most people can be authenticated via a fingerprint scan. A determination of L for retina collectability means that there are issues surrounding collecting retina scans that stop it from being more highly collectable.

	Characteristic						
Biometric	**Universality**	**Distinctiveness**	**Permanence**	**Collectability**	**Performance**	**Acceptability**	**Circumvention**
Fingerprint	H	H	M	M	M-L	H	H
Palmprint	H	H	M	M	M	M	M
Retina	H	H	H	L	M	M	L
Iris	H	H	H	M	H	M	L
Face	H	M	M	H	M	M	L

Table 2.1: Characteristics of selected physiological biometrics. The determinations of High (H), Medium (M) and Low (L) are the author's interpretations.

Some of the issues with physiological biometrics, particularly collectability and circumvention, are addressed by the use of behavioral biometrics. Basing a biometric pattern on a person's actions may be less distinctive, but they often do not require additional hardware and may be gathered while the subject engages in normal activities rather than requiring them to submit to a scan.

2.6.2 Behavioral Biometrics

Physiological biometrics can be limiting, particularly in a mobile device environment, because they can require additional hardware to gather the biometric. Behavioral biometrics are

known to be less distinctive than physiological biometrics [65], but have several benefits over physiological systems. They are easy to gather while the subject goes about other tasks and thus are ideal for transparent authentication. Their collection does not usually require special hardware and thus may be more cost effective. Much research has been undertaken into various possibilities, including keystroke dynamics [19, 66–68], speaker verification [69–71], touch screen interaction patterns [72] and device use patterns [55, 73]. The list below briefly describes key behavioral biometrics.

Signature: is the distinctive way in which a person signs their name. Signature metrics include the writing instrument pressure (electronic signatures only), shape of letters and other additions such as dots and flourishes. It has long been accepted as a method of identification and verification by government and legal bodies as well as by the general public. It requires use of a writing instrument and either a paper or electronic surface upon which to sign, which are relatively low-cost. Signatures are highly susceptible to forgery, although signature verification by a person or improved pattern matching algorithms can improve these methods [74, 75]. Signatures can be highly variable and thus require acceptance within certain tolerances rather than exact matches.

Gait: is the characteristic way in which a person walks. Gait is a complex biometric because it combines spacial and temporal issues, in that both movement in a 3D space as well as the timing of each movement must be measured [76]. It uses hardware such as accelerometers and gyroscopes for measurement [77], which are common in mobile devices, as well as 3D cameras. Issues with gait include limited universality since those who cannot walk are immediately exempt, which includes young children, the elderly and infirm and people in wheelchairs. Gait may also vary depending on the subject's weight, age and mental state, among others, and thus is not highly invariant [78]. It is computationally-intensive both in feature vector creation and matching due to the complexity of the data gathered.

Device Use: attempts to gather patterns in how subjects use devices such as desktop and laptop computers and mobile phones. Examples of device use include sequences of events, use of shortcuts versus menu items, and routes taken while walking or driving [79]. These patterns, which can be gathered from such things as browser history and application notifications, are expected to be moderately distinctive, and require a relatively long training period [80]. They are subject to variability due to device changes (i.e., if the subject begins to use a new mobile phone), and changes to the functionality of the device (i.e., new software or programs on the computer). This sort of monitoring may be cause for concern in subjects due to its similarity to eavesdropping.

Typing: The way a person types is expected to be distinctive and is known as *keystroke dynamics* [81]. Measurements of the speed, frequency of characters and n-grams as well as the pressure with which keys are pressed are gathered and combined into a distinctive pattern [18, 82]. It is considered discriminatory for verification but not identification [59]. This biometric is highly variable due to mental state, subject position (i.e., standing, sitting or walking) and keyboard familiarity. It can be gathered using a standard keyboard while the subject goes about other tasks. Privacy issues include fear of keylogging. Keystroke dynamics may be subject to replay and imitation attacks, although the latter may be more difficult.

Voice: is both a physiological and a behavioral biometric. The physiological aspects include measurements of voice features that change due to the distinctive shape of the subject's features such as larynx, glottal folds, mouth and lips [59]. The behavioral aspects include pronunciation, word frequency and use and accent. The physiological aspects are relatively invariant over a person's life, but the behavioral aspects may be affected by mood, state of mind, age and medical conditions such as the common cold. Voice biometrics are not very distinctive and unsuitable for large-scale deployment due to issues with contamination from other noise during recording [83]. It is subject to misuse due to recording and replaying a subject's voice, and it can be gathered without the subject's knowledge.

Table 2.2 shows the biometrics described above in relation to Jain's seven biometric characteristics. As with the physiological biometrics discussed in the previous section, a determination of High (H), Medium (M) or Low (L) denotes how well the biometric adheres the characteristic. The individual determinations are based on the author's opinions and knowledge of biometrics.

	Characteristic						
Biometric	**Universality**	**Distinctiveness**	**Permanence**	**Collectability**	**Performance**	**Acceptability**	**Circumvention**
Signature	H	M	L	H	M	H	M
Gait	L	M	L	H	M	H	L
Device Use	L	M	L	H	M	L	L
Typing	H	M	L	H	M	M	M
Voice	H	M	M	H	M	H	M

Table 2.2: Characteristics of selected behavioral biometrics. The determination of High (H), Medium (M) and Low (L) are based on the author's opinions.

The biometric chosen should reflect expected use in the regular context. The current research in mobile device authentication has examined these, and other, behavioral biometrics. However, studying the literature in the wider field of authentication may provide a basis for new research in mobile authentication.

In 2008, Yampolskiy and Govindaraju published a survey paper on behavioral biometrics [65], with the intent of gathering different types of behavioral biometrics and outlining studies that have been based on them. The biometrics included programming style and "soft" behavioral biometrics such as word knowledge and mathematical ability, and biometrics such as keystroke dynamics and gait analysis. Yampolskiy and Govindaraju's work provides a basis for selecting behavioral biometrics to use for a particular purpose.

While the 2002 study by Clarke *et al.* [18] showed the applicability of various biometrics to mobile device environments, their determination of such applicability was based on devices available in 2002. Other researchers have since attempted to use the increasingly feature-rich mobile devices developed since that time to test a wide range of behavioral biometrics including speaker recognition, signatures and handwriting, touching and tapping and device use patterns. In 2004, Gamboa and Fred published a study on using captured taps via a pointing device (a mouse) [55], and while their study does not specifically refer to mobile devices, their method has the potential of being adapted to touch-based mobile devices since they gather the clicked location itself rather than relying on what specifically performed the click (e.g., finger or mouse). The main contribution of Gamboa and Fred's research is in the biometric feature selection decisions, which are the specific parts of a biometric pattern that make it distinctive from those of others; these are the parts that are presented to the pattern classifier.

One interesting advantage that touch–screen mobile devices have is that touches have been shown to contain potentially distinctive patterns, although it does not deliver the level of assurance required for authentication. Frank *et al.* performed a study ($N = 41$) to examine the applicability of screen touches as a behavioral biometric for use in a continuous authentication system [72]. Their study resulted in misclassification error rates in the range of 4%, which although quite low, is not low enough to support authentication unless it is combined with another biometric.

Many other behavioral biometrics have been considered for authentication. Examples include device use patterns (also called service utilization) [12, 56, 79, 80, 84], signatures and structured writing [85, 86], gestures and gait [87] and mouse movement [88–90]. The common thread running through all of these methods is that there is much uniquely identifying information in behavioral biometrics, and these methods lend themselves to a transparent authentication method.

While many behavioral biometrics are suitable for authentication purposes, a few are particularly applicable to mobile devices. This platform is characterized by short, frequent interactions [7, 8, 91], and has several input modalities such as a keyboard, touch screen and microphone. A number of characteristics of behavioral biometrics are particularly important in choosing a biometric to use on a mobile device:

1. They can be gathered without using extra hardware on the device;
2. They can be gathered while the user goes about his or her normal device use;
3. They have the potential to be processed on the device itself;
4. They may be sufficiently discriminatory in terms of ability to verify that a user is the owner of the device.

Therefore, biometrics such as keystroke dynamics and speaker verification, among others, that use the existing device hardware are worth examining for their applicability as authenticators on a mobile device. To this end, the focus now moves to examining the current research into two behavioral biometrics, keystroke dynamics and speaker verification, to the mobile device environment. These biometrics have been selected because they can be gathered transparently on mobile devices while the user goes about regular tasks. Whether they meet the third and fourth characteristics is one subject of this research.

Keystroke Dynamics

Keystroke dynamics has a long history as a potential authenticator, beginning with Spillane's seminal work in 1975 [92]. Interest in keystroke dynamics has grown over the years and now includes several patents [93, 94]. Early studies focused on desktop and laptop computers, although more recent work has extended to mobile device keyboards of all type [95–103]. These studies vary in the types of metrics used, the pattern classification technique, the data collection modality (fixed or free text), and how much information must be collected before error rates are low enough to support authentication.

Several studies have focused on mobile keyboards [68, 104, 105] although few have been performed on a soft keyboard device in which the keyboard is displayed on the device screen and the user types by tapping the displayed characters [106]. Huang *et al.* [107] studied keystroke dynamics on a soft keyboard but their results were based on removing the data from the mobile device for processing with a customized pattern classifier, and uses fixed text in the form of a username and password. Many of these studies required modifications to the device such as removing the keyboard and wiring it to a desktop computer for processing or

using a virtual keyboard created with infrared scanners [108]. This may have had an impact on the study participant's typing patterns.

In a similar study, Clarke *et al.* [19] performed a keystroke dynamics study on a thumb-based keyboard using text messages and number entry. They also removed the keyboard from the mobile device and attached it to a desktop for processing, which may have changed the user's interaction with the device. It is unclear, however, whether their results would generalize to all thumb-based keyboards that were used in their natural state.

Current keystroke dynamics work on mobile devices use a variety of metrics including key hold-time, inter-key latency, finger pressure and number of errors while typing [109]. Finger pressure, in particular, produced a high level of accuracy [106], but required adaptations to the device screen that may inhibit widespread use of this metric. Many studies use two or more metrics to reduce the authentication error rate due to the inadequacy of any one metric. Karatzouni and Clarke found that key hold-time is unreliable for use with mobile devices that have full QWERTY keyboards [20], thus the practice of combining metrics may have value.

In 2009, Hwang *et al.* [82] proposed a keystroke dynamics authentication method that used fixed-text modality. They found that only five short (4 character) typing samples were needed in order to accurately identify the device owner. Their low error rates with such a small amount of data were due to a unique metric – they asked users to type with an artificial rhythm that was encouraged through the use of audible cues while typing the fixed text. Hwang *et al.*'s work is another example of *password hardening* [99, 110–112], in which the point-of-entry protection provided by a password or PIN is enhanced by adding the use of keystroke dynamics while the password or PIN is typed.

In addition to varying the number and type of metrics used, current research also varies the type of pattern classifier used to match a gathered keystroke pattern to an existing pattern. Options include, but are not limited to, neural networks [19, 20, 106], Support Vector Machines [96], fuzzy classifiers [104], and various distance measure such as Mahalanobis, Euclidean, and Manhattan distances [68, 82, 113]. Clarke and Furnell compared the accuracy and speed of both statistical and neural network classifiers [19] for classifying keystroke dynamics data. They found that neural networks had lower misclassification errors, but higher processing requirements when compared to statistical classifiers. Along the same lines, Haidar *et al.* used fuzzy classifiers, neural networks, and statistical classifiers on the the same data to create a user profile that was used in combination with the password typed to identify the user [114]. This research compares several statistical classifiers, including k-nearest neighbor, decision trees, and Naïve Bayes approaches. The result of their work was that the statistical classifiers had fewer misclassification errors than the neural networks and fuzzy classifiers. For all classifiers, they found that combining classifier output reduced

the error rate further, although evidence of this is not given in their paper.

A concern with the current research in keystroke dynamics on mobile devices is with the validity of the findings given the experimental procedures used. These studies often change the way in which the user interacts with the device; mobile devices are often used while standing, walking and sitting, for example, where desktop and laptop computers are generally used while seated. These changes may alter the user's typing patterns and thus do not represent their natural patterns. These studies fail to take into account this difference, which means that the reported error rates for the study may be artificially low.

Speaker Verification

This section introduces the current literature on speaker verification as a behavioral biometric capable of supporting authentication on a mobile device. Since speaker verification is strongly related to other speech-related biometrics such as speaker diarization [115, 116] and speaker identification [117, 118], other types of similar work will be included here. The main difference between speaker verification when compared to other voice biometrics is the application for which the results are used (i.e., verification rather than identification).

In 1999, Li *et al.* investigated whether speaker verification methods were sufficiently discriminatory for use in an authentication system [70]. They examined the speech patterns of 100 speakers making long-distance telephone calls. They found that they could distinguish a particular speaker with error rates between 1.8% and 2.6%, and related this to a confidence measure for how confident the authentication system was that the speaker was who they claimed to be. Li *et al.*'s study is similar to that performed for this dissertation work. Similarities exist in the modality (telephone conversations) and the use of speaker verification methods rather than speaker identification. Li *et al.* chose to use 8 kHz audio samples, which they claim to be a standard for voice verification. Their work is important in terms of defining the terms and methods of speaker verification in general. However, their claim that "The performances of lab data indicate that both systems are ready for real-world deployment" [70] cannot be supported with the small amount of lab data they chose to provide in their study, especially since the statistical significance of the values that are reported have not been calculated.

Speaker verification and other related voice biometrics are not a panacea for authentication. There are serious limitations in the technology, as argued by Bonastre *et al.* in 2003 [83]. The authors warn that, as of the time of the published paper, "there is no scientific process that enables one to uniquely characterize a persons voice or to identify with absolute certainty an individual from his or her voice" [83]. The warnings presented by Bonastre *et al.* show that extremely low error rates for speaker verification are not likely within the confines of (relatively) current state-of-the-art in the field.

In 2010, Kinnunen and Li reported on text-independent speaker recognition systems [119]. Like the contribution of Li *et al.* [70], they too provide an important look at the key terms and technologies used in text-independent speaker verification systems. Unlike Li *et al.*, though, Kinnunen and Li did not perform an experimental study; their goal is to compare commonly-used technologies. They also provide definitions for text-dependent and text-independent studies, define significant problems with voice-based identification and provide speaker selection and speaker models. Most significantly, however, this paper suggests the definitions for cooperative and uncooperative speakers (i.e., those who know their speech is being sampled and allow it and those who do not).

In 2006, O'Gorman *et al.* [120] developed an authentication method that uses a spoken PIN that can be uttered in front of an attacker without revealing the secret, and also a Query Directed Password [120] that the authors considered more memorable than a regular password. These directions in authentication research explore methods to manage the problem of password memorability and reuse. Furthermore, the authentication methods they propose are intended to be used in busy locations where secret knowledge leaks are common.

It is possible that the method described by O'Gorman *et al.* could be extended for use in transparent authentication. For instance, the password or PIN could be a particular combination of words or phrases that the device owner speaks frequently in normal conversation, although this would have to be tested against others who also speak them to determine whether they are distinctive enough to use as an authenticator, particularly if the words or phrases are commonly used. Finally, O'Gorman *et al.*'s speaker authentication methods are static since the answers required of the pre-selected questions must be the same as those provided during enrolment.

In 1994, Gish and Schmidt [118] performed two speaker verification studies that used the SWITCHBOARD corpus as a datasource. SWITCHBOARD was created by Texas Instruments [121] and consists of about 2500 long-distance telephone conversations on landlines rather than mobiles from around 500 speakers. It was created to provide a source of training and testing data for many speech processing algorithms, specifically for speaker verification research [121]. The purpose of Gish and Schmidt's two experiments was to determine whether noisy channels and different channels (e.g., handsets) make a significant difference to the pattern classification of these samples. Their results showed that it was significantly harder to perform a high-quality, low error rate pattern classification when the channel being tested was not represented in the training set. This result is significant because it led to further research on how to improve pattern classification techniques to allow for these differences.

There is sufficient evidence that behavioral biometrics on their own may not be sufficiently distinctive to deliver a confident assertion of the identity of the user of a mobile device. Con-

sideration should be given to combining biometrics to provide sufficient evidence of identity. This allows exploration of the strengths of different biometrics and whether combining them can offset their weaknesses.

2.6.3 Multimodal Biometrics

Multimodal biometrics are intended to minimize the weaknesses of individual biometrics by providing more information upon which biometric decisions can be based. They are versatile tools since there are several possible aspects from which their combination can be considered. Some possible methods for combining biometrics are as follows [57]:

1. Measuring the same trait with multiple sensors:

   (a) Single trait, multiple sensors (i.e., one finger, multiple fingerprint scanners used in succession). This combination method gathers the same biometric pattern with a series of scanners. The two options at this point are to combine the features of the individual scans into a single feature vector, which is then presented to the pattern classifier. This type of *feature* fusion technique is often characterized by too many features, making the pattern computationally difficult to classify. Another method is to process the raw data from each scanner into separate feature vectors and present each to a pattern classifier. Then, the scores from each are combined into a single score; this is an example of *score-level* fusion.

   (b) Single trait, multiple classifiers (i.e., present the same fingerprint to more than one pattern classifier and aggregate the results of each into a single decision) This differs from the above method in that only one biometric scan is taken. The raw data from the scan is converted into a feature vector that is then presented to a series of pattern classifiers, each of which outputs a score for the input biometric. These individual results are then combined into a single score upon which a decision can be made.

   (c) Single trait, multiple versions (i.e., more than one finger, both irises or both retinas, etc). In this combination method, two or more scans of the same *type* of biometric are taken, but the source of the biometric is different. For instance, a fingerprint of the subject's index finger and thumb may be taken in two successive scans, and presented separately to pattern classifiers. Then, the scores from the pattern classifiers can be combined into a single score upon which a decision can be made.

   In each of these cases, an alternative to combining scores is to combine *decisions*, as described below.

2. Measuring more than one distinct biometric identifier and combining the results of individual pattern classification [122, 123]:

   (a) **Feature Fusion:** combines biometric features from two or more different biometric patterns by concatenating extracted features into a single larger feature set, which is then presented to the pattern classifier. The two biometrics should be independent of one another; that is, varying one should not result in variations in the other. The new feature vector has a higher dimensionality and should result in a more reliable biometric decision, particularly if a pre–processing step is used to select the most distinctive features from each biometric.

   (b) **Match Score Fusion:** two or more biometrics are presented to pattern classifiers and are assigned a score (but not a decision) that identifies how close the gathered feature vector is to the template vector. The scores are then combined and a decision is made based on the combined scores.

   (c) **Decision Score Fusion:** multiple biometric feature vectors are presented to pattern classifiers, and are placed into one of two groups: accept or reject based on the output of the pattern classifier. The individual accept and reject scores are then combined, often with a weighting factor, to output a single accept or reject decision. The decision combination can be made based on a majority voting scheme, such as the one described by Zuev and Ivanov [124].

Each of these combination methods has pros and cons, and selection should be made based on the needs and user base of the system under development. One consideration when making a selection is whether the system will perform identification or verification, as defined in Section 2.6. In general, if a claim of identity is made previously, verification is the correct mode. For instance, a person using a mobile device may be considered to inherently claim the identity of the device owner; in this case, verification rather than identification is performed.

Research into multimodal biometrics was undertaken as part of this work because behavioral biometrics are not expected to provide sufficiently low error rates for authentication [59]. As the research highlighted in this section shows, combining two or more biometrics may improve the error rates when compared to a single biometric. There is a large body of multimodal biometric research that uses variations of common biometrics, such as fingerprints, facial recognition and ear shape. This section focuses on research that uses biometric combination methods to reduce error rates.

Iwano *et al.* [125] use a combination of speech patterns and ear shape to authenticate mobile device owners. They selected these two biometrics because voice patterns have many issues such as noisy environments that make this biometric error-prone, but ear shape is relatively

static and thus can be used to increase robustness. Iwano *et al.* contaminated their audio samples with white noise in order to test the improvements multimodal biometrics provided. They found that combining the two biometrics reduced the error rate from approximately 38% for the individual biometrics to just over 10% for their combination. Iwano *et al.* did not experiment with the possible transparency of ear shape by taking the ear image while a call was made from the device, likely because devices available at the time of their experiment did not routinely have a camera on the side of the phone that is held to the ear.

In similar work to Iwano *et al.*, Rokita *et al.* [126] used a mobile device camera to take photos of users' hand and face. They extracted similar features from each photo to create a single feature vector. They did not compare the error rates of individual and combined biometrics, but found that there is a point where adding more features resulted in a higher error rate. One issue in Rokita *et al.*'s work is that it is unlikely that the photo pre-processing they perform prior to pattern classification, as well as the classification itself, will be performed on a mobile device due to processor speed and memory limitations. Furthermore, even if the device were capable of this, the amount of time that passes between taking the photos and an authentication decision is likely prohibitive.

In 2005, Fierrez-Aguilar *et al.* [74] published a comparison of fusion techniques for multimodal biometrics that was based on the quality of the metric at the time it was gathered. Their study used fingerprints and signatures as the biometrics. Their proposed fusion techniques, while not specific to mobile devices, may be used in a mobile environment. Their technique determined the quality of the biometric at the time it was gathered, and used it to influence the result of the pattern classification by adding a quality factor to the classification formulae. One of the major issues in Fierrez-Aguilar *et al.*'s work is that the quality of the fingerprints is determined by a human expert as part of their experiment. This was not of concern during their work as they used a corpus of fingerprints that had already been examined for quality. This, however, would not be viable in a production environment, particularly on a mobile device.

Multimodal biometric authentication methods based on facial recognition and voice biometrics are common choices [127–132]. Poh and Korczak [132] developed a text-dependent voice biometric, which differs from the text-independent method used for this thesis work. The use of facial recognition implies that the authentication method is explicit, much like the work of Rokita *et al.* and Iwano *et al.* Poh and Korczak attempted to simplify the head positioning requirements typical to facial recognition systems by only using features around the eyes of the image, and accept that they will lose important features for the sake of simplicity. In addition, the voice analysis uses a spoken password for comparison purposes; this supports the determination that this was an explicit authentication method. Such text-dependency represents a limitation in their work since speaking a password gives attackers a strong advantage, as does the fact that the password is quite short (3 seconds of record-

ing time) since then it may be easier to spoof the intended owner. Poh and Korczak's work shows that the fusion of these two biometrics improves the error rates seen when using one biometric, which supports the use of multiple biometrics in authentication.

An important consideration when deploying an authentication mechanism is whether the user base for which it is intended will accept and trust the method. To justify the use of biometrics as an authenticator, research has been performed to determine user acceptance of it.

2.6.4 User Acceptance of Biometrics

In 2001, Clarke *et al.* [41] undertook a study to determine the security needs of mobile device owners, and what types of security precautions they take or would be willing to employ if they were available. Their study found that while many users did not utilize the PIN available on their mobile devices, they would be willing to consider using biometrics as an authenticator. The authors opine that the reason for the participants' apparent acceptance of some biometrics and rejection of PINs may have to do with whether the participant had heard of the biometric in question. For instance, the authors state that many of the participants were likely to have heard of fingerprints, but were less likely to have been exposed to the idea of ear geometry as a biometric. Therefore, they conclude, the participants may have been responding to knowledge of the method rather than their desire to use it as an authenticator.

In a follow-up study, Clarke, Furnell and Reynolds [12] examined several physiological and behavioral biometrics with the goal of identifying those that were viable in a mobile device context. Their study examined the physiological biometrics fingerprints, facial recognition, and iris scanning; the behavioral biometrics they focused on were voiceprints, signatures, keystroke dynamics, and service utilization. Their study found that the latter two behavioral biometrics were the most viable on mobile devices because they do not require additional hardware and can be sampled transparently. They chose not to study physiological biometrics further due to the cost and difficulty of applying the required additional hardware such as fingerprint scanners to the mobile device environment. Therefore, it is worth considering behavioral biometrics in more detail.

While acceptance is an important aspect to biometric authentication techniques, performance of the method must also be taken into consideration. The methods for reporting the distinctive ability of biometrics vary, but have been the subject of significant research efforts.

2.6.5 Biometrics Performance Metrics

There is currently no widely accepted method for reporting the results of biometric studies. Crawford [133] provided a step towards such formalization by reviewing keystroke dynamics research on mobile and non-mobile platforms and proposing a list of reportable statistics. Many studies in this area continue to report a dizzying array of error rates and performance curves. This makes it challenging to compare study results, and has the potential for drawing incorrect conclusions.

Pattern classifier output is sensitive to many factors, including algorithm choice, amount of training data, the chosen features in the feature vector, and within-participant variation. These factors will have an effect on the performance metrics computed for each classifier. Within the current literature, there are several widely-used methods to report the quality of a particular pattern classifier; those described here are used to report the results of the biometrics studies in this research. Table 2.3 shows the different types of metrics that can be considered for any pattern classifier. It shows all of the possible results in a two-class problem, with the class decisions made by the classifier in the columns, and the true, known classes in the rows. The diagonal from top left to bottom right shows the number of correctly classified patterns. *True accept* and *true reject* are seen when the classifier produces the same result as the known classification for the pattern. *False accept* and *false reject* are when the classifier produces the opposite result to the known classification. Many studies report both false accept and false reject rates but do not report the true accept and true reject rates. In many research papers, these values are known as *false (or true) positive* and *false (or true) negative*, but the terms false (or true) accept and reject will be used in this research.

		Predicted Class	
Actual Class		**Positive**	**Negative**
	Positive	True Accept	False Reject
	Negative	False Accept	True Reject

Table 2.3: A generic confusion matrix for a two-class decision problem.

Several different types of error rates are commonly reported in biometrics studies. There is some disagreement in the research community as to which rates are important [133], but the generally accepted errors are as follows (see Figure 2.2):

Crude Accuracy (CA): Also called *misclassification error*, this standard method of reporting results is simply the number of incorrect classifications made when comparing the classifier output to the known true class for the pattern [134]. As it is a combination of the next two metrics, it delivers minimal value on its own.

False Accept Rate (FAR): Also called *Type I error* or *false positive*. FAR expresses the

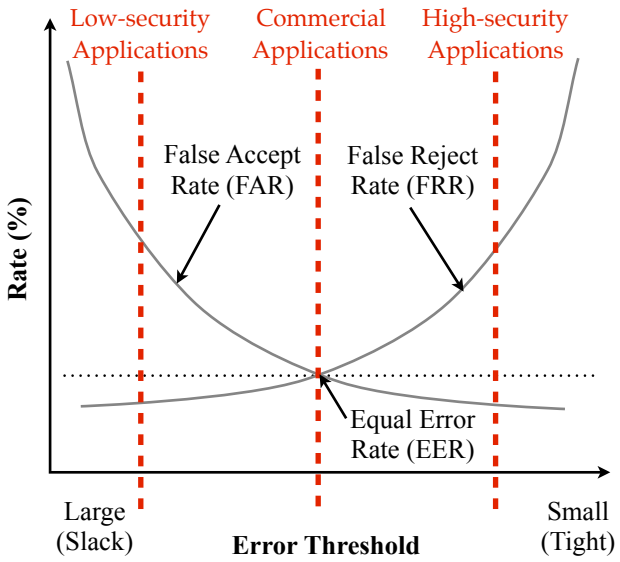


Figure 2.2: Generic classifier performance metrics, with threshold levels for secure, insecure, and unknown security levels, showing the relationship between EER, FAR and FRR. These curves do not represent results from this or any research. Adapted from [12].

likelihood that an unauthorized user (i.e., an impostor) will be granted access to the protected resource. High FAR values are often seen as a significant problem because they represent an intrusion into a protected system, although the determination of a threshold accepted level is left to particular implementations. Let FA be the number of false accepts and NI be the number of impostor patterns. FAR is calculated as in Equation 2.1 [135]:

$$FAR = \frac{FA}{NI} \tag{2.1}$$

False Reject Rate (FRR): Also called *Type II error* or *false negative*. FRR represents the likelihood that an authorized user will be denied access to the protected resource. It can be seen as an annoyance to the authorized user since it means that they will have to attempt to reauthenticate, perhaps more than once. Let FR represent the number of false rejects from the classifier output and NA be the number of authorized user patterns. Then, FRR is calculated using Equation 2.2.

$$FRR = \frac{FR}{NA} \tag{2.2}$$

The relationship between FAR and FRR has been described as mutually exclusive since it is impossible to both reject and accept the same authentication attempt [18, 136]. While such a statement is true, care must be taken when using such a description for these two related error rates. FAR and FRR also share an inverse relationship – it should not be assumed from the use of the term "mutually exclusive" that no relationship exists between the error rates.

The proof of such a relationship lies in the definition of Equal Error Rate (EER).

Equal Error Rate (EER): EER is defined as the point at which the plotted curves of FAR and FRR values cross [12], as seen in Figure 2.2. In this figure, a large or "slack" error threshold means that the value above which an authentication attempt is granted access is low. In other words, more authentication attempts will be accepted than with a small, or "tight", error threshold. The terms "small" and "large" in this context refer to the range in which accepted attempts reside. With small error thresholds, the range of values that are accepted is small, and the reverse for a large error threshold. EER can also be determined by plotting the ROC curve for the classifier, as detailed below, and determining its abscissa by plotting a diagonal line from the upper left to the lower right corners and observing where the two lines cross.

ROC Curve: A Receiver Operating Characteristic (ROC) curve, as seen in Figure 2.3, shows the relationship between FAR and True Accept Rate (TAR), which is the number of patterns that actually belong to the positive class [134]. The ROC curve shows the overall usefulness of the results of the pattern classification. The closer the line comes to the upper left corner of the graph, the better the method is at correctly identifying or verifying users. Furthermore, since this curve is based over all thresholds, it can be used to select a viable threshold at which the classifier in question is most accurate.

Area Under Curve (AUC): AUC is a measurement of the area under the ROC curve [137] for a given classifier and a given user. It is a representation of the probability of a true response (either positive or negative) when classifying data – a random classifier will have an AUC value of 0.5 (50%) and an ideal classifier will have an AUC of 1.0 (100%). AUC is a summary that attempts to represent the entire ROC curve in one value. As such, AUC calculation loses some information and nuances of the original curve since the individual tradeoff values that make up the curve are lost.

The European Standard for Access Control Systems (EN 50133-1) states that a biometric authentication system must have a False Accept Rate (FAR) of less than 0.001% and a False Reject Rate (FRR) of less than 1% in order to be used in production systems [138]. However, the error rates suggested in EN 50133-1 are not specific to *behavioral* biometrics, which are known to be less distinctive than physiological biometrics [65]. Therefore, the values stated in EN 50133-1 may not provide a suitable benchmark to use in determining the applicability of any behavioral biometric. Instead, the error rates of related work in the particular biometric field may be used as a benchmark.

The research presented to this point has examined biometrics for use on mobile devices, including user acceptance of them and the methods of reporting their errors. Their use in the mobile device environment has the potential to provide a continuous, transparent authentication method. This concept has been studied by several researchers, both for mobile device

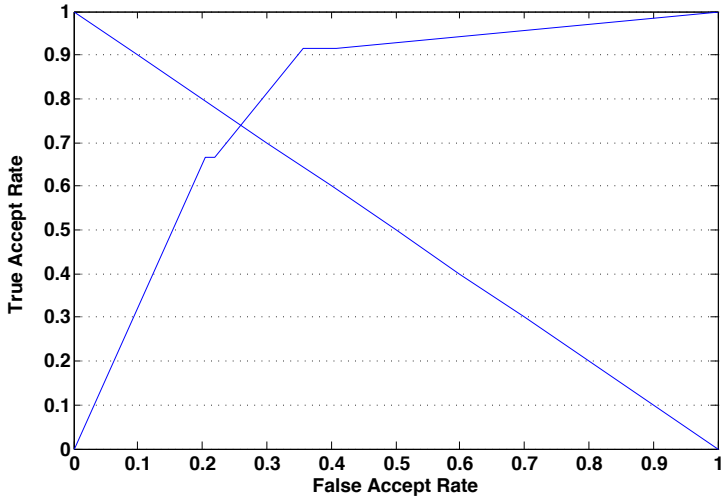


Figure 2.3: An example ROC curve. The AUC for this curve is 80.13%. The EER (25.99%) is the point where the two lines cross.

and standard computer authentication.

2.7 Transparent and Continuous Authentication

In the context of this work, transparent authentication is defined as verifying the identity of the user of a mobile device without explicitly requiring user effort. This section discusses current literature on transparent authentication, which is also called implicit [56, 80, 139] or zero-interaction authentication [140]. For the purposes of this research, the term transparent authentication will be used in place of all other similar terms. The implication is that the data upon which to base an authentication decision is found in *how* a person uses a device while that person goes about their regular tasks.

Attempts to describe transparent authentication have been seen as early as 2002. Corner and Noble [140] describe a system that uses a physical hardware token worn by the user that wirelessly communicates a master password to a laptop that is then used to unlock a larger password file. However, their system requires the user to explicitly authenticate to the token, which means that their system is not truly transparent, although the authors state that the owner would have to authenticate less often than with regular password systems [140].

Since early attempts in 2002, the idea of transparent authentication has been researched heavily. In 2008, Briggs and Olivier [141] suggested creating "biometric daemons" that learned their owners' behavior, pined and eventually died in the absence of this behavior. Their ideas were based on the *His Dark Materials* trilogy of books by Philip Pullman [142] in which a human is matched with a small creature, called his daemon, who essentially represents the human's soul and will pine and die in the absence of the human, as will the

human in the absence of the daemon. Briggs and Olivier paint a picture of an electronic form of daemon that requires imprinting and nurturing – essentially equivalent to the more traditional enrolment and testing phases of authentication systems. This is a rather fanciful embodiment of the idea of transparent authentication, and Briggs and Olivier do not go beyond the thought-experiment phase.

In 2009, Tanviruzzaman *et al.* [52] experimented with the idea of an "electronic pet" that learns the owner's patterns to determine the identity of the person using it. Tanviruzzaman *et al.* go beyond Briggs and Olivier's idea to describe which biometrics they intend to gather: gait, location, voice, fingerprints, and facial patterns. It is unclear how they intend to combined these processor-intensive biometrics into a single decision. Their system is described as "silent and less intrusive most of the time, i.e., the authentication process runs continuously in the background until a higher level of security is needed" [52], which implies that it is an transparent authentication system although they do not use such a term specifically. They have extended Briggs and Olivier's thought experiment but have not yet, to this author's knowledge, empirically evaluated their proposals.

Marsh and Briggs [143] expanded the idea of a biometric daemon further by developing a concept they call *device comfort*. Rather than focusing solely on the capabilities of the biometric daemon, Marsh and Briggs examine the device's relationship with its owner – device comfort – and relate it to the concept of trust in an electronic environment. Marsh and Briggs consider electronic trust to be related to the device's knowledge of its owner's patterns, preferences and data on the device, although this is a simplification of a increasingly well-defined concept [144]. In this way, the concepts of electronic device trust may also be used as a foundation for transparent authentication on a mobile device. While trust in an electronic environment is applicable to transparent authentication, it is not a topic that will be addressed in this dissertation. The reader is directed towards more thorough sources for further information [144, 145].

Marsh and Briggs point out that device comfort is not the same as the owner's relationship with the device. Device comfort is from the point of view of the *device* rather than from its owner. The owner's point of view is achieved through customizing the device interface. Device comfort originates from the point of view of the device since it is the device that does the majority of the work. Their work was intended to fill the gap between increasing mobile device functionality and the limited amount of security on such devices. In this way, the work of Marsh and Briggs has similar goals as the work described in this dissertation, although their work does not reference authentication. They do, however, provide a "roadmap" that can be used to apply the concept of device comfort to a mobile environment, which allows transparent authentication on mobile devices to be developed with this in mind as well as with a strong idea of how electronic trust applies.

Riva *et al.* have proposed a solution to mobile device authentication [146]. Their solution, called *progressive authentication*, does not appear to be a new method; instead, it is an examination of whether *when* we decide to authenticate (as opposed to *how*) improves the security of mobile devices. The authors suggest that if the mobile device remains in the owner's possession, even if the owner is not currently interacting with it, there is no need to authenticate should the owner decide to use the device. However, if the device has lost contact with its owner (i.e., it is lent to someone else or placed on a surface), then authentication should be required. Riva *et al.* suggest a hybrid transparent authentication method since with enough information regarding who possesses the device, authentication is transparent. Otherwise, explicit authentication is required. Their user study showed that they were able to reduce the number of required authentications by 42%.

Shi *et al.* cite the frustration owners experience in entering standard passwords and PINs into small mobile device keyboards as a motivating factor for transparent authentication [80], a notion that is supported by Allen *et al.* [147]. Shi *et al.* proposed using a smartphone to record the owner's location, motion, phone call patterns, and application usage to determine whether the owner is in possession of the smartphone, and allow him or her to access the Internet based on this determination. Shi *et al.* have the main goal of using sources of identifying data that do not require typing since they see smartphone keyboards as frustrating and error-prone. The high levels of reported success show that device use patterns are indeed a rich source of information.

A behavioral biometric that may be used transparently was studied by Conti *et al.* [148] in 2011. Their work uses the accelerometers and gyroscopes in mobile devices to track the movements a user makes when answering a phone call, with the purpose of using these distinctive movements to identify the device's owner. Their research showed that there is sufficient uniqueness in this pattern to verify the owner's identity, and suggest that it can be used in an transparent manner. The gap in the research that Conti *et al.* have filled is that passwords and PINs on mobile devices are not required when answering an incoming phone call, and thus do not protect the device from misuse in that case.

The applicability of behavioral biometrics to transparent authentication has been presented in this section. While these individual instances of studies to determine their feasibility are important, what is required to solve the mobile device authentication problem is a model that is biometric and platform independent, so that as technology progresses, the model stays useful.

2.8 Transparent Authentication Frameworks

As transparent authentication becomes a more heavily researched field, authors are beginning to note that a framework for supporting such an authentication method would be invaluable. Such a framework removes the focus from specific methods, such as particular biometrics that can be gathered while the owner goes about other tasks, and moves on to the requirements to support device use and functionality choices in light of the specific biometric decisions. Since it is a relatively new research aim, there have been only a few published attempts at creating such a framework or model.

In 2008, Furnell, Clarke and Karatzouni [149] described a framework called Non-Intrusive Continuous Authentication (NICA). NICA was designed to allow transparent, continuous authentication that provides more security than secret-knowledge techniques such as passwords and PINs. They used keystroke dynamics, facial recognition and voice patterns as their biometric identifiers. NICA includes the idea of an "alert level" that relates the biometric decisions to the device functionality, and also uses an explicit authentication method when the transparent method is insufficient.

In 2009, Clarke, Furnell and Karatzouni [17] published an extension of their 2008 NICA paper [149] that moves beyond the framework stage to lab experiments and testing. Their study used two approaches to assess user perceptions of current and potential future forms of authentication. The first approach was to issue an online survey that gathered 297 respondents. The second approach was a focus group with 12 participants. The result of both approaches was a determination of a series of stakeholder requirements for a transparent authentication framework, that they then used to improve the NICA framework. Once complete, they built a prototype of NICA and performed an end-user trial with 27 participants. The overall result was that 92% of the trial participants felt that NICA provided a more secure environment compared to traditional authentication methods, although the traditional methods were not specified.

Clarke, Furnell and Karatzouni's work depends on having a server–side extension to the mobile device that manages authentication decisions, data synchronization and biometric profiling. This implies that the data must leave the device for processing, which may have privacy implications. Furthermore, Clarke, Furnell, and Karatzouni used a mocked-up device that existed as an interface on a desktop computer due to limitations in mobile device memory and programmability at the time of their experiments. This requires a certain "suspension of disbelief" on the part of the study participants since they are not using the device itself, which may affect the outcome of the study. Advances in technology have made it possible to extend Clarke, Furnell and Karatzouni's framework to allow for on-device processing.

Jakobsson *et al.* [139] were among the first authors to use the term "implicit authentication".

In their work, Jakobsson *et al.* further filled the transparent authentication gap by proposing a model for it, although they did not create a working implementation of it. They suggest two important findings: first, that both authentication security and usability are increased by using transparent authentication when compared to explicit methods; second, that while transparent authentication can be used on many devices, it is ideally suited to mobile devices because of their access to a rich source of data about the device owner [139]. They did not, however, proffer justification for either claim. Jakobsson *et al.* also consider data that is available from the carrier and from off the device itself. They chose to use the call frequency as the sole biometric, although they do make reference to other biometrics such as keystroke dynamics that could be used.

The model that resulted from Jakobsson *et al.*'s work is yet another step in the direction towards realizing transparent authentication but has issues such as how the results of call frequency calculations relate to what may be done on the mobile device. Another issue in their work is that there is no comparison of the strength of call frequency to other biometrics that can be used on a mobile device.

In 2007, Mazhelis and Puuronen [79] produced a framework for *user substitution detection (USD)* on mobile devices. They claim to consider security "detective" methods, rather than "preventive" aspect of security. The difference with these two approaches is that the former investigates who committed an intrusion after it happens, while the latter focuses on preventing the intrusion from happening. Mazhelis and Puuronen claim that USD is only closely related to authentication rather than true authentication. In their view, true authentication can be assumed to end when the user is granted access to a resource, although more contemporary definitions of authentication refer to allowing or disallowing the use of specific resources, services and data. Mazhelis and Puuronen's work focuses on a strong psychological connection between the user's actions and their uniqueness and classification ability in place of implementation methods and pattern classification techniques.

Biometric techniques and the frameworks that support them require methods for comparing gathered patterns to known patterns. Typically, machine learning techniques such as statistical and neural network-based pattern classification algorithms have been used for this purpose.

2.9 Pattern Classification and Machine Learning

The pattern matching tasks in most biometrics research use standard pattern classification algorithms to make decisions. Figure 2.4 shows a typical workflow for a pattern classification task. The workflow begins with the selection of one or more classifiers for the data at hand. Next, the classifiers are trained with a subset of the gathered data to create a model to which

test data and future patterns will be compared. Once training is complete, the classifier model is tested for accuracy by presenting the trained classifier with some test data that was not used in the training phase. The results of the tests are then examined and measurements such as those described in Section 2.6.5 are generated.

The next step is to simplify the model by identifying those data features that provide the most discriminatory information to the classifier and removing those that provide minimal information. This *reduction of dimensionality* simplifies the classifier since there are fewer features to compare during any one classification task. It may be the case that no features may be removed if they are all equally important.

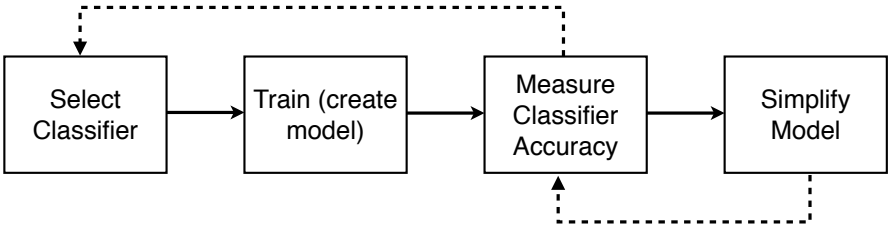


Figure 2.4: Generic workflow for a pattern classification problem.

If the error rates are not sufficiently low, then a different classifier is selected, as shown by the top dotted arrow in the figure. If the dimensionality can be reduced, then the new model is once again tested with the same data as in the previous test and the error rates are compared to see if simplifying the model had an effect. The expected outcome is that the error rates are no lower than previously, although there are some cases where simplifying the model may increase the classifier's accuracy. Once the most accurate classifier has been chosen, the model can be used to classify new data.

The classifiers used for this research are suitable for use on a mobile device platform, which has limitations in memory, battery life and processor speed. For this reason, several factors influence classifier choice, as follows:

Simplicity: the classifier should have a simple algorithm.

Speed: the classifier should make a decision within a few seconds so that device functionality is not impaired.

Accuracy: the classifier should, within the constraints of behavioral biometric accuracy, have acceptably high AUC and low EER, FAR and FRR values. The exact definition of "acceptable" relies on the specific classifier and biometric implementation.

The following classifiers, all of which meet the first two requirements stated above, are commonly used and are standard in some programming language libraries:

Naïve Bayes (Gaussian and Kernel Density): This probabilistic classifier uses either a Gaussian or Kernel Density Estimation, and requires data independence within each class. Data independence means that the presence of one feature in the data is unrelated to the presence of any other feature. Such a characteristic lends itself well to small datasets since using additional features does not require an exponentially larger dataset. The specific types of Naïve Bayes classifier tested were Gaussian distributions for estimation, and also a kernel density model since the data in this study does not follow a Gaussian distribution. A kernel density Naïve Bayes classifier does not make assumptions regarding the distribution of the data to be classified (a Gaussian model assumes a Gaussian data distribution), and is suitable for continuous rather than discrete measurements. Based on the training data, the probability density of the timing features for each class is estimated using a kernel function. When new data is presented to the classifier, it is placed in the class whose estimated density function gives the highest value for the new data [150].

Decision Tree (DT): This classifier is often used to map decisions used to place data into one or more classes. Each node in the tree represents a feature in the data that can be used to determine to which class it belongs. The leaf nodes of the tree represent the classes. Decision trees are a suitable classifier for the data in this research because they are fast to classify new data, and have misclassification error rates that are comparable to more complex classifiers. They also make no assumptions about the data's distribution.

k-Nearest Neighbor (k-NN): This algorithm creates a feature space by plotting all training data on an n-dimensional graph as single points. When new data is classified, the data point is plotted on the same graph, and then assigned the majority class of its k nearest neighbors, where k is a parameter that can be adjusted by the experimenter. Smaller values of k allow classification when there is only a small amount of training data. Manhattan and Euclidean distance measures were tested with the k-NN pattern classifier to determine whether the distance measure used makes a difference to the accuracy of k-NN.

The following two classification algorithms were considered for use with this research, but were discounted because they did not meet one or more of the above requirements:

Support Vector Machine (SVM): This classifier is commonly used with two-class problems that use supervised learning methods. The model represents the data as points in space that are divided by a hyperplane; one of the two classes is on each side of the hyperplane. New data is classified by plotting it in the same space and predicting its class based on which side of the hyperplane the point falls. While Support Vector

Machines are well-suited to the type of data seen in this study, they are often slow to classify and require significant processor speed and memory. For these reasons, SVM was considered a poor choice for this work and was thus not tested.

Neural Network (NN): Artificial neural networks are based on the network of biological neurons that are present in the human brain. They consist of a series of artificial neurons or nodes that are interconnected in such a way that they can be used to model complex relationships between the network's inputs and outputs. One of their uses is to find patterns in data. A basic neural network consists of at least three layers: the input, output and hidden layers. The nodes in each layer are connected to each other in that each node in each layer passes its output to each node in the next layer, as shown in Figure 2.5. The interconnections may be weighted, and the training (or learning) phase of a neural network updates the weights for each interconnection. Generally, neural networks have high accuracy but are slow to train and to classify. Furthermore, they may require large amounts of training data, depending on the application. For these reasons, they were considered unsuitable for this work and were not tested.

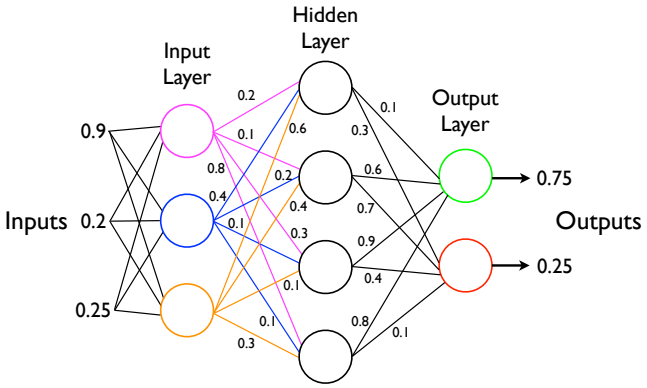


Figure 2.5: Example of a neural network. The numbers on each interconnection are the weights for that connection. They are updated during training.

Pattern classifiers are used to make decisions about the class of new data based on the features and known class of previously provided data. They may be used for pattern matching in biometric implementations, but should be accurate, algorithmically simple and provide timely answers to new classification data.

2.10 The Transparent Authentication Framework

The current research into authentication, specifically that which takes place on a mobile device, has provided a basis for further work in the field. Specifically, the research examined

in this chapter has provided evidence to support the need for a mobile device authentication method that has the following properties:

1. Allows provision of authentication and access control methods that go beyond point-of entry security, such as those provided by passwords and PINs.
2. Works transparently to remove the need for explicit authentication, therefore providing less intrusive and frustrating approach to authentication. This is intended to be suitable for use with the short-but-frequent patterns that characterize mobile device use.
3. Takes into account user needs and wishes for mobile device authentication.
4. Maintains the device owner's privacy by keeping their personally identifying data on the device.

To address these needs, this research introduces the Transparent Authentication Framework. This framework provides the basis for creating a continuous, transparent authentication method for mobile devices by sampling behavioral biometrics while the owner uses their device in a normal manner. The behavioral biometrics, which can be combined into a multimodal biometric in order to potentially decrease errors, are used to make decisions about the device user's identity, which is then used to affect *device confidence*. This concept is defined as the certainty that the device has that its user is the device owner. It is linked to device service provision by mapping the device confidence to allowable tasks and data on the device. If the minimum device confidence for a particular task or data is higher than the currently calculated device confidence, the task or data is not accessible. The authentication provided by such a framework is continuous in that the user's behavioral biometrics are sampled each time they use the device, and it is transparent in that the device owner does not have to explicitly authenticate each time they wish to use the device.

2.11 Terminology Used in this Dissertation

The terminology used in the remainder of this dissertation is defined here. This section is not exhaustive; many definitions are provided as they are used.

Mobile device: a typical smartphone. One definition of a mobile device is "any device that can send and receive telephony services, contains a SIM card, and is controlled by a mobile network operator" [151]. However, as stated by Becher *et al.*, this definition is too broad since it also includes mobile phones that are not considered smartphones – those with limited functionality beyond telephony and text messaging. These latter type of mobile device are called *feature phones*, and are not included in this definition

of mobile device. Furthermore, laptop and desktop computers and tablets are excluded as well. Therefore, the use of the term *mobile device* should be seen as synonymous with the term *smartphone*. Specifically excluded from this definition are laptop and desktop computers, e-book readers, and tablet computers.

Identification: determining whether a person is amongst a group of authorized users of a protected resource.

Verification: determining the veracity of a claimed identity.

Keystroke dynamics: a behavioral biometric that uses typing patterns to identify or verify the identity of the typist.

Speaker verification: a behavioral biometric that uses a person's speech patterns to verify a claimed identity.

Multimodal biometric: a combination of two or more biometric identifiers into a single biometric, with which a decision on identity verification can be based.

2.12 Summary

This chapter has introduced the fundamental concepts that form the basis of this research. It has provided a high–level view of authentication, biometrics and pattern classification, and has defined key terms that will be used throughout this research. The purpose, therefore, has been to provide a strong basis upon which the state-of-the-art research in the field of authentication may be built. As such, this chapter has also examined the current research in the field of mobile device authentication, with specific focus on biometrics and transparent authentication. Current work on several physiological and behavioral biometrics was examined, with a focus on how these biometrics relate to authentication on mobile devices. Specific use of behavioral biometrics as transparent authenticators on mobile devices was examined; finally, other frameworks for transparent authentication were reviewed. The findings of the current research in the field of mobile device authentication has provided a basis for the Transparent Authentication Framework presented in this dissertation. This Framework expands upon the current state-of-the-art by providing continuous, transparent mobile device authentication based on behavioral biometrics. It is this Framework that addresses the research question and resultant hypotheses presented in Chapter 1, and thus provides novel work in the field of authentication.

Chapter 3

Transparent Authentication Framework for Mobile Devices

This chapter introduces the main contribution of this research: the Transparent Authentication Framework. The Framework provides a model for designing and developing a transparent authentication mechanism to verify the user's identity on a mobile device; its target audience is mobile device developers. It is intended to be conceptually device and operating system neutral, whether manufacturer or version.

The sections in this chapter cover concepts of *device confidence* (the level of certainty that the current user is, in fact, the device owner), the data required as input to and the processes associated with the Framework, and the requirements for the biometrics that may be included. Each section contains a discussion of the rationale for the inclusion or concept, major elements thereof, and examples of what technology may be used. Since the Framework is intended to be device and software independent, it provides a basis for selecting the best available design choices for provision of a continuous, transparent authentication method on a variety of mobile devices.

3.1 Framework Overview

The Transparent Authentication Framework provides a model for creating an authentication mechanism for mobile devices that goes beyond point-of-entry secret knowledge-based tools such as passwords and PINs. The Framework describes an authentication model that utilizes measurable patterns of device use that can be gathered during its normal functioning. The data collected during execution of these common tasks and the identifiable patterns within them are used to verify the identity of the owner of a mobile device. Presumably, owner verification can also enable access to data and functions. In terms of the three access control

components introduced in Chapter 2, the Transparent Authentication Framework exists in the convergence of something you have, something you know and something you are. The placement of the Framework in the greater access control field is shown in Figure 3.1.

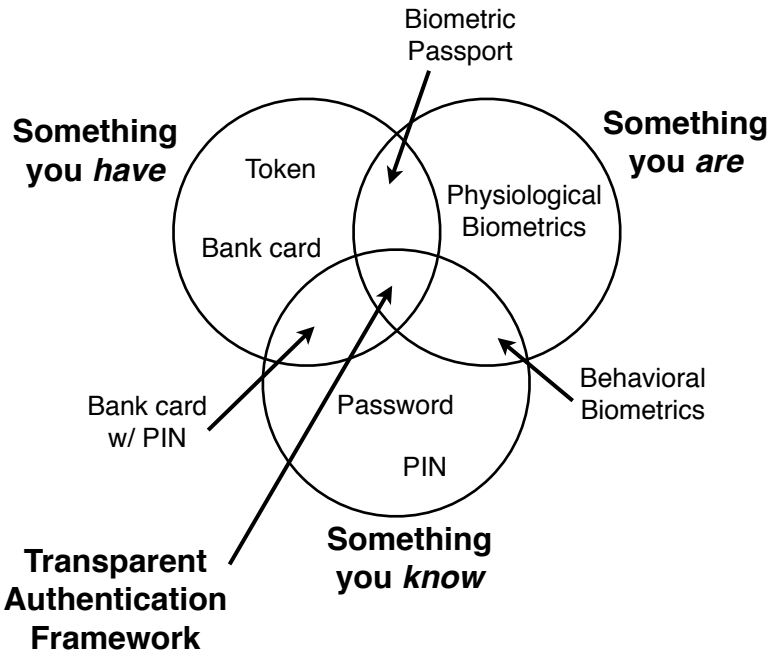


Figure 3.1: Placement of the Transparent Authentication Framework in the access control domain. It resides at the convergence of the three standard access control factors: something you have, something you know and something you are.

This chapter introduces the concept of *device confidence* as a means of expressing the ongoing confidence that the current user is also its owner. Device confidence increases and decreases in response to biometric matches and non-matches. This continually changing measure is mapped to on-device tasks and data. *Task confidence* is the level set by the owner as the minimum threshold at which access to the task is permitted so that those considered private or sensitive require a higher device confidence to be accessed. The intention is to provide a more nuanced approach to security when compared to binary allowed/not allowed security currently provided by passwords and PINs.

The authentication delivered by this Framework is *transparent* in that it does not require explicit user interaction. Instead, it takes advantage of uniquely identifying behavioral features available while the owner uses the device. The authentication is also *continuous* in that it is updated even when the device is not being used. In this way, it goes beyond traditional point–of–entry authentication provided by passwords and PINs, which only protect the device up to the point the secret knowledge is entered. The transparent, continuous nature of this Framework supports flexible, dynamic authentication.

In some situations, the device may have too low a confidence to permit access to functions

requested by the owner; for instance, if the owner has only recently begun using the device. To manage these situations, device confidence can be augmented with secondary, explicit authentication methods such as challenge questions, a PIN or password. This secondary method is not all-access; a correct challenge response raises device confidence by increments. This means that access to sensitive information remains possible only at the highest device confidence levels.

The Transparent Authentication Framework addresses implementation concerns for mobile device methods. Such concerns include the required characteristics of included behavioral biometrics, how to combine these into a multimodal biometric for additional security, and allowing the device owner to customize options. The Framework includes a process that maps biometric decisions to device confidence, as well as the types of biometrics and classifiers that may be used. Furthermore, this Framework respects device owner privacy since it is designed in such a way that all device owner data remains on the device rather than being processed at a server that then delivers a biometric decision.

The Transparent Authentication Framework uses device confidence, as calculated by biometric decisions, to determine what tasks may be completed or data may be accessed. Each task or data is assigned a confidence level either by default or explicitly by the device owner. If the device confidence is greater than or equal to the required task/data confidence, then the device user is allowed to complete the task or access the data. Otherwise, the task or data access is denied and the user must attempt to raise the device confidence. This general flow is shown in Figure 3.2.

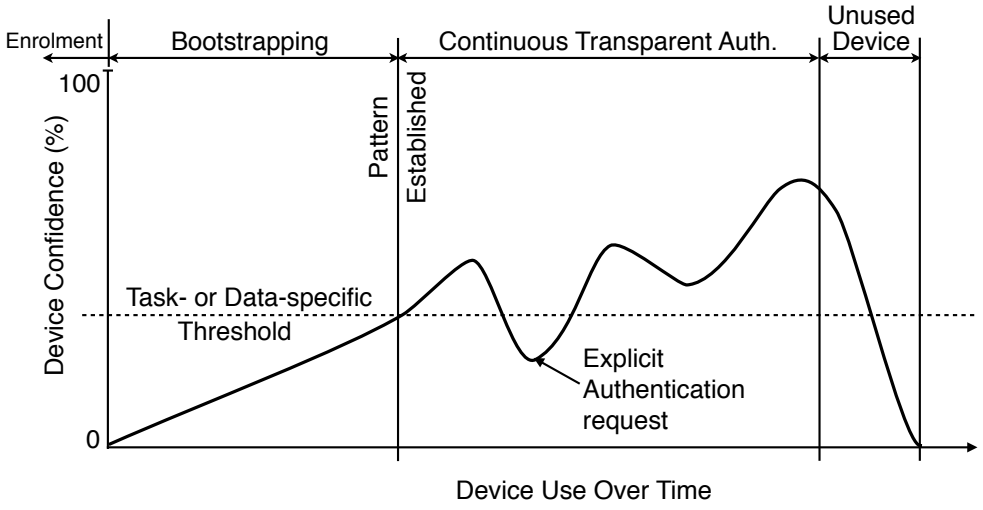


Figure 3.2: Transparent Authentication Framework general flow.

The workflow begins with the device confidence at 0% since the device owner has not yet provided biometrics with which a baseline can be established. The Enrolment phase, seen at the upper left corner of Figure 3.2, precedes the Bootstrapping phase, and allows the owner to provide biometric samples and set initial task confidence levels. Using the Enrolment phase

in this manner allows device confidence to start at a level higher than 0%. This determination, and the level at which device confidence will start, is left to the developer who uses the Framework as a model for an application. As the device begins to know its owner, the device confidence raises until the owner pattern is established. Until this point, access to data and tasks is limited. During the Continuous, Transparent Authentication phase, the device user is allowed access to a task or to data if their current device confidence is above the threshold set for that task or data. During this period, the device confidence fluctuates as biometric matches and non-matches occur. At some point, the device user has requested a task or data whose task confidence is higher than that currently seen on the device. The owner may then enter their response to the explicit authentication challenge, which may increase the device confidence to a high enough level to allow the task or data access.

If the device is no longer being used, the device confidence lowers until it reaches 0% since there is no biometric or explicit authentication data upon which to base increases or maintain confidence. This regular degradation in device confidence implies that the device will eventually disallow access to all tasks and data. This can be overridden in the event the device is once again used by entering the explicit authentication response. Then the flow shown in Figure 3.2 begins anew at the beginning of the Bootstrapping phase.

3.2 Device Confidence

Device confidence is defined as the certainty the device has that its current user is the device owner. This confidence is altered by the results of biometric matching – a non-match lowers the device confidence and a match raises it. Device confidence can also change based on a correct or incorrect response to an explicit authentication method that is used as a supplemental security mechanism. As such, the device confidence can be thought of as a value ranging from 0 to 100, where 0 means that the device either does not recognize the current user at all, or that there is not enough information to determine who is using the device, and 100 means that the device is fully confident that the device owner is currently using the device. These values can be seen as percentages, meaning that a device confidence of 50 is half as confident as a device confidence of 100. Device confidence begins at 0% for new, untrained devices.

3.3 Data Structures

The Transparent Authentication Framework has several types of data structures that support it. The data structures are designed to be flexible and to use commonly available data primitives that are available in most programming languages. They minimize the amount and type

of data required in order to support on–device processing, which helps protect privacy by keeping the device owner's personal information on their device.

3.3.1 Event Objects

Event objects are created from raw biometric patterns or from correct answers to explicit authentication requests. There are two different types of event objects: biometric and explicit authentication. The event object e_{ij} refers to the j-th instance of authenticator i, where $i, j \in \mathbb{N}$. All event objects, e_{ij}, are tuples that consist of the following:

- A timestamp, t;
- A feature vector, fv, that is a representation of either the biometric or the answer to the explicit authentication method;
- A probability, p_i, that the feature vector belongs to the device owner.

$$e_{ij} = (t, fv, p_i)$$

The two different types of event objects, while the same in structure, have a few differences in the contents of each variable in the tuple, as shown in Table 3.1.

Event Object	t	fv	p_i
Biometric	Time the biometric was gathered	Represents the gathered biometric. Different for each biometric	Probability that the biometric belongs to device owner based on fv
Explicit Authentication	Time the explicit authentication was gathered	User-supplied answer to the explicit authentication	$p_i \in \{0, 1\}$

Table 3.1: The differing components of each of the two types of event objects.

Event objects are the smallest building block of the Transparent Authentication Framework. They are the basis for all decision–making processes within the Framework, and should be organized in such a way that older event objects are considered before newer. Buffers are one example of a data structure that supports this requirement.

3.3.2 Input Event Object Buffers

The input event object buffers, as pictured in Figure 3.3, hold the event object instances created from the raw biometrics. There is one input event object buffer for each type of biometric used in the Framework and one additional buffer for the explicit authentication input. As each biometric is determined to belong to the device owner, it is placed in the corresponding sample buffer until needed to recalculate device confidence. The buffers are First In First Out (FIFO), meaning that the oldest samples will be replaced by newer samples as they become available.

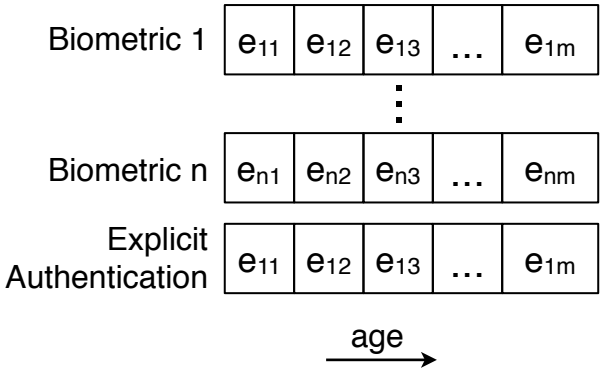


Figure 3.3: Input Event Object buffers. There is one buffer for each type of biometric and one for explicit authentication.

Figure 3.3 shows each buffer as having length m, but in practice it is likely that buffers would have different lengths. The recommended data structure for input event object buffers is a FIFO queue, allowing older samples to be discarded as new samples are added.

3.3.3 Training Event Object Buffers

Training event object buffers follow the same basic structure as input event object buffers, except that there are only buffers for individual biometrics. Explicit authentication buffers are not required because this authentication method does not rely on pattern classification and thus does not require periodic retraining. In the case of multimodal biometrics, the pattern classification is performed on an individual biometric basis. The individual biometric buffers are populated with those event objects that have been used to raise device confidence and that are considered particularly descriptive of the mobile device owner, and are used on a first–come, first–served basis. Therefore, the recommended data structure for training event object buffers is a FIFO queue.

3.3.4 Device Confidence Value

The device confidence is a value that ranges between 0 and 100, as described in Section 3.2. Since device confidence is a value that is compared to other similar values, the recommended data type is a floating–point number.

3.4 Processes

There are several processes that use the data and data structures described in the previous section. The processes comprise the Framework's main actions throughout the lifecycle described in Section 3.1. Each process in the following sections is introduced by stating its inputs and outputs and then describing its intended use.

3.4.1 Update Biometric Input Event Object Buffer

Input: A raw biometric pattern and a trained classifier for that biometric

Output: A new biometric event object and a newly updated biometric event object buffer

This process updates the event object buffer for a given biometric pattern. It begins when a biometric included in the Framework is gathered. A single biometric event is defined using a windowing process. Each raw biometric pattern is gathered for a set period of time, after which a new event object is created. The time period depends on the type of biometric being gathered, and thus is an optional parameter within the Framework. All of the raw biometric data gathered during this set time is considered one pattern, and is processed into a feature vector that is then presented to the trained classifier for the particular type of biometric. The classifier output is a probability that the device user is its owner, p_i for the i-th pattern.

After classification has taken place, an event object as described in Section 3.3.1 is created and added to the buffer that corresponds to the biometric used in the calculations. The new event object remains in the buffer until it is replaced by a newer event object, or is culled due to its age.

3.4.2 Update Explicit Authentication Event Object Buffer

Input: A response to the explicit authentication challenge

Output: An explicit authentication event object and a newly updated explicit authentication event object buffer

When the device owner answers their explicit authentication challenge, a new explicit authentication event object is created. Its time t is the time the answer was provided, and in place of a feature vector, the answer to the challenge is provided. These event objects are created whether or not the correct response is given; in the case of an incorrect response, the probability is set at 0%. For a correct response, it is set to 100%. In this way, a correct answer has a high weighting when calculating device confidence, which is expected since explicit authentication is similar to knowing a password, which is commonly used as the sole check of a person's identity. Once the new event object has been created, it is added to the end of the explicit authentication event object buffer, in much the same way as for the biometrics buffers.

3.4.3 Compute Averaged Biometric Probability

Input: An event object buffer

Output: A single probability that is the combination of the input probabilities in the buffer

Possible methods for creating a multimodal biometric are described in detail in Chapter 6, which introduces a study used to determine whether multimodal biometrics are more distinctive than individual biometrics. Within the Framework, multimodal biometrics are computed by combining several probabilities using the methods detailed in Chapter 6.

The probability of the gathered biometric belonging to the device owner $P(e)$ is based on the output of the biometric classifier (p_i), and is subject to a weighting factor w_i:

$$P(e) = w_i * p_i$$

The weighting factor w_i is used to make some classifications more substantial than others, in that they are considered more reliable (i.e., more representative of the device owner) and thus should have more effect on device confidence. For instance, if the biometric is known to be relatively indistinct for a given user, its weighting should reduce the biometric's overall effect. w_i represents the weighting factor for the age of biometric pattern i (which could be multimodal), and is calculated as follows:

$$w_i = \begin{cases} 0 & \text{if } a_i \geq 30 \text{ days} \\ 0.5 & \text{if } 7 \text{ days} \leq a_i < 30 \text{ days} \\ 1 & \text{if } a_i < 7 \text{ days} \end{cases}$$

where a_i, the age of the biometric in days is calculated as follows (now represents the current date and time):

$$a_i = now - t_i$$

For example, if the current time (now) is July 13, 2012 at 9:45am (this is the biometric's t value in the event object) is subtracted from a speaker verification biometric gathered on July 12, 2012 at 2:45pm, the difference is 1101 minutes, which is approximately 0.76 days, calculated as follows:

$$\frac{1101 \text{ minutes}}{\left(24\frac{\text{hours}}{\text{day}} * 60\frac{\text{min}}{\text{hour}}\right)} \cong 0.76 \text{ days}$$

and thus $a_i \cong 0.76$ days in the calculation of w_i, above.

The values of 30 and 7 days used in the calculation of w_i can be adjusted to suit the specific instance developed using this Framework. This weighting factor accounts for aging biometric samples that may not adequately represent the device owner's current patterns. For example, if the device owner rarely makes phone calls on the device, it is possible that there may not be many voice event objects in the voice biometrics buffer. It is also likely that any input events in that buffer may be old compared to other buffers since the voice buffer would not receive enough input to refresh the buffer frequently. In this case, the age of the event object would mean that it will hold less power to affect device confidence due to this weighting factor. This is unrelated to the probability combination methods in Chapter 6 since the weightings described above are intended to be applied to the posterior probabilities output by the pattern classifier on an individual biometric basis. The methods in Chapter 6 are more robust and intended for multiple biometric instances.

3.4.4 Compute Device Confidence

Input: The multimodal event object(s) and $P(e)$

Output: Updated device confidence

The calculation for device confidence takes into account the probability $P(e)$ that all the biometrics currently residing in the event object buffers belong to the device owner. $P(e)$ is used to calculate the new device confidence, C'_d, as follows:

$$C'_d = \begin{cases} C_d + value & \text{if } P(e) \geq t_{owner} \\ C_d - value & \text{if } P(e) < t_{owner} \end{cases}$$

where $value$ is a developer–determined percentage that is added to C_d if $P(e)$ is greater than or equal to the threshold, t_{owner} and subtracted from C_d if $P(e)$ is less than t_{owner}.

The threshold t_{owner} is a developer-chosen value that represents the point above which the biometric is considered to belong to the device owner. The developer chooses $value$ based on the sensitivity of the system, and its tolerance for false accepts and rejects. For example, $value$ could equal 1% in a system that is very sensitive, such as a work–related mobile device used by a government official, but be raised to, say, 10% for a personal mobile device. It is also possible that the developer can change $value$ depending on what stage the authentication system is in currently. In particular, a higher $value$ amount could be used during the training stage to reduce the amount of time required before the device can be used in a regular manner, and then lowered for regular device use.

Device confidence should be up to date at all times since it is what allows or disallows access to data and tasks. Two methods of determining when to recalculate device confidence are suggested:

1. On Demand: The current C_d value is compared to the level assigned to the requested task (C_t) and found to be too low to allow the task. As seen in Figure 3.6a, two possible actions result from this comparison: if the current device confidence is greater than or equal to the task confidence, the user is allowed to begin the task. If the current device confidence is less than task confidence, then the user is prompted to enter their explicit authentication challenge. If they answer it, the update explicit authentication event object buffer process begins, which adds a new event object to the explicit authentication buffer that is then used to recalculate C_d. At this point, the newly calculated device confidence is compared to the required task confidence, and the cycle begins again. If the user does not enter the explicit authentication challenge response, then there is no information with which to recalculate C_d, and the task is not allowed.

2. Periodically: The Transparent Authentication Framework can access the biometric decisions via the event objects in each buffer on a timed basis. A default value may be set (every 10 minutes is one suggestion, although the default timing should be verified after testing on a production system). The suggested timing can be adjusted by the device owner in the enrolment stage. This periodic recalculation of device confidence is how continuous authentication is provided, since the device confidence is recalculated even when the device owner is not interacting with the device. As seen in Figure 3.6b, this recalculation uses event objects from the individual biometric event buffers, if any are available. The figure refers to this as *recalculation* of device confidence rather than raising it since it is possible that the biometrics in the buffer may not represent the device owner. In such a case, device confidence may be lowered during recalculation. If no biometric samples are available, then the device confidence is lowered according to the calculations given in the previous section. During the initial training phase of the Framework, the period for recalculation is set to zero,

which means that as soon as a new event object is created, it is used to increase the device confidence immediately.

When device confidence is recalculated, it takes into account all event objects in the biometrics event buffers. The process begins by taking the event objects in all biometric buffers and computing a combined biometric probability for each type of biometric, as shown in the top of Figure 3.4. This calculation provides a single probability that the user is the device owner that is based on all current event objects for that type of biometric. After this is completed for all biometrics, the composite probabilities are then combined into a single probability using the biometric combination method, as seen in the lower half of Figure 3.4. The result is a single combined probability that is based on all event objects in all buffers. Since explicit authentication event objects are also represented as probabilities, they can be included in this process in the way described here. This single probability is then used to recalculate device confidence.

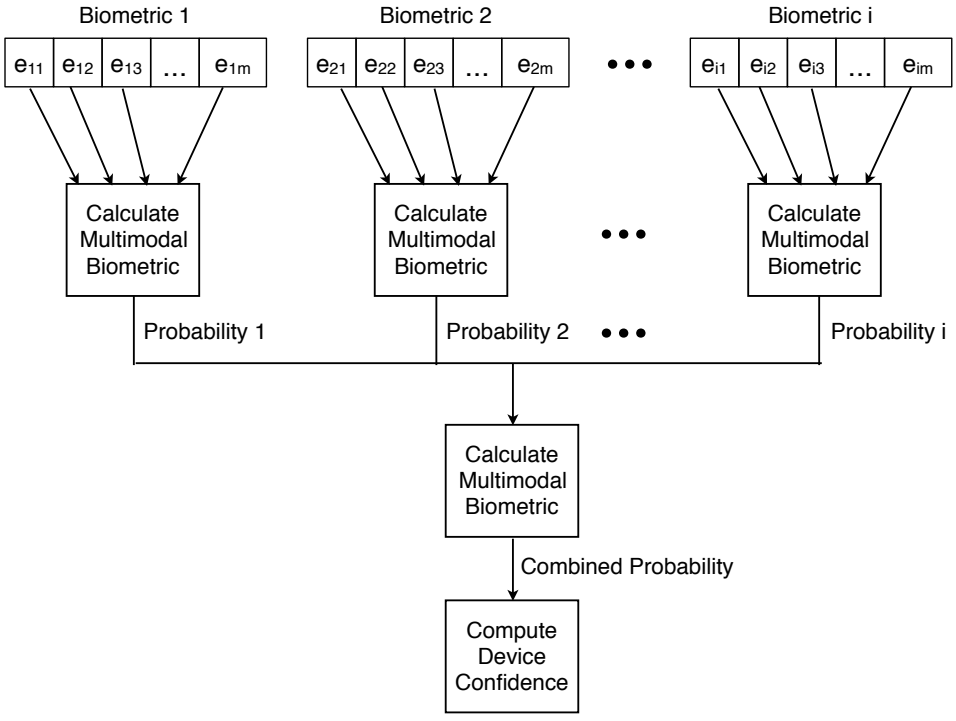


Figure 3.4: Multimodal biometric calculations to update device confidence.

3.4.5 Make Task Decision

Input: Current device confidence, required task confidence

Output: Binary value that represents whether the user can perform the task (1 is yes; 0 is no)

This process is wholly dependent on the current level of device confidence. There is a mapping between the current profile's device confidence, C_d, and the tasks that are possible. The task confidence, C_t is determined by the device owner during enrolment, although default values should be set initially by the developer who uses the Framework. Figure 3.5 shows an example mapping of several tasks to device confidence levels. It shows that as device confidence increases, the user is allowed access to tasks of increasing security or sensitivity level (None, Low, Medium and High).

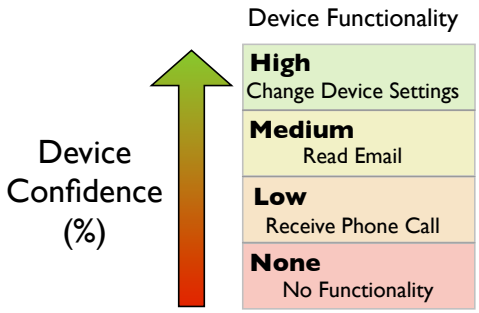
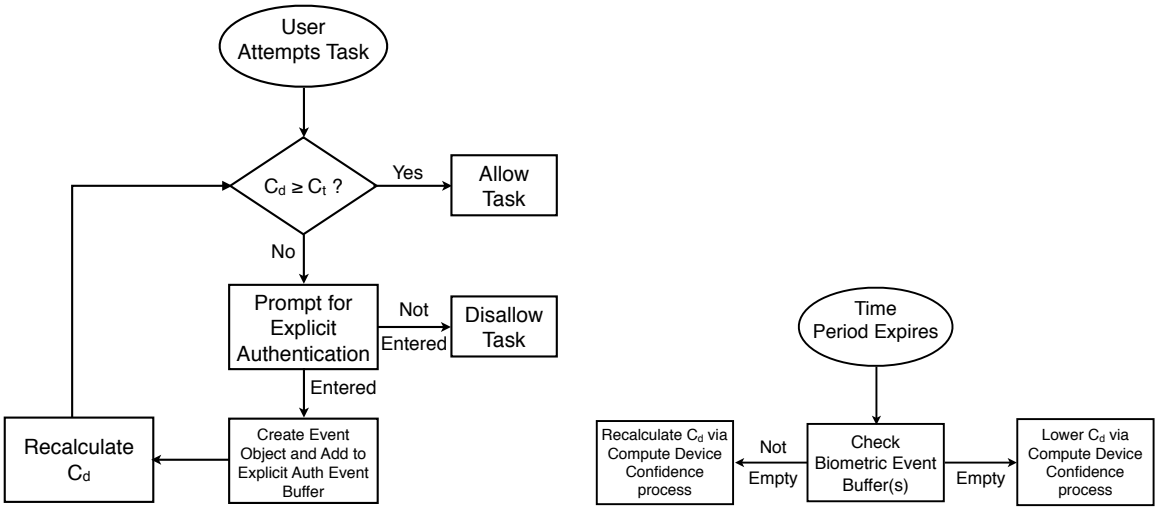


Figure 3.5: Mapping of device confidence to task or data threshold. The tasks in each security level are examples only.

The decision process for allowing or disallowing a task is shown in Figure 3.6.



(a) On-demand device confidence recalculation and task access.

(b) Periodic device confidence recalculation.

Figure 3.6: The two methods by which device confidence is recalculated. The On-Demand method shows task access logic.

3.4.6 Update Training Event Object Buffer

Input: Biometric event object(s) and a threshold for minimum acceptable event object probability

Output: A new training event object buffer

As each biometric event object is used to recalculate device confidence, it is also used as input to the process that updates the training event object buffer for that biometric. The developer selects a threshold above which the probability, p_i, in the event object is considered high enough to adequately represent an owner pattern. This threshold is important because it must be high enough that excellent representations of owner behavior are used for retraining, but poor representations are discarded. One suggestion for the threshold is 75% as this is likely a sufficiently high probability to ensure quality of pattern match.

As each event object is found to exceed the chosen threshold, it is added to the end of the training event object buffer. There is a different buffer for each biometric used in the Framework, although there are no buffers for explicit authentication since it does not require a trained classifier.

3.4.7 Refresh Buffers

Input: The buffer to be updated, and the age factor above which a biometric is removed (biometric dependent)

Output: The updated buffer

As new event objects are added to the buffers, older event objects are removed from the buffer if the buffer is full when the new object is added. This allows for newer objects to be used in device confidence calculation. However, this method of refreshing data does not allow for the case where the buffer is not full, which may occur when few biometric patterns are gathered (e.g., when the device owner speaks infrequently). In this case, the buffers are refreshed by periodically assessing the age of each event object in the buffer and removing those that are older than a given age. This age can be selected by developers since some applications may allow older event objects, particularly those with a low security requirement.

3.4.8 (Re)train Classifier

Input: Training event object buffer

Output: A retrained classifier model

The classifiers used to make biometric probability decisions must be trained on known owner patterns prior to regular use in the Transparent Authentication Framework. The initial training is a special case of the periodic retraining described below. The only difference is that the training set is populated either from user–supplied patterns gathered during Enrolment or Bootstrapping. Otherwise, the process is the same as for retraining the classifier.

The classifiers require periodic retraining because the owner's patterns will change over time due to many factors such as device familiarity, varying device and use patterns. Always comparing to old patterns as trained by the old model will mean that there will be more false negatives over time as the owner's patterns develop.

There are several possibilities for policies that govern retraining classifiers:

1. **Periodic:** Regular retraining based on time (i.e., every 24 hours).

2. **Training set deviation:** If the probabilities assigned to newer biometric patterns have declined (especially if they decline in a gradual, regular way) then this could signal that retraining is necessary since it is possible that the device owner's patterns have altered legitimately. If there is a sudden or dramatic difference in a few patterns, these would not be included in the retraining set because they would not exceed the threshold set for inclusion in the training buffer.

3. **Device Idleness:** If the device is not used for a long period of time, the data used to train the classifiers is likely less useful because of its age. Retraining should take place from either the Enrolment or Bootstrapping stage to provide up–to–date information to the classifiers.

4. **Processor Idleness:** This can signal a time when the device is not being explicitly used (for example, when the owner is sleeping, assuming it is not being used by an unauthorized person) and may mark a time when retraining is least likely to impact device use. This is important because retraining a classifier may not always be simple or quick. This policy can also be combined with the regular updates idea since it would make sense to do a regular update when the device owner is usually known to sleep, which is periodic.

The device and processor idleness policy suggestions can be combined with device situational awareness to minimize the impact of retraining, in terms of memory and battery life use. Situational awareness can include the device's current location as well as whether the device is currently connected to a power source. In the former case, the device may choose not to retrain the classifiers if it is in a place it does not frequently habitate. This concept is related to Marsh and Brigg's idea of *device comfort*, in which the device may disallow tasks if it is in an unfamiliar location or using an unsecured Internet connection [143]. In the latter

Figure 3.7: Biometric lifecycle.

case, battery life is preserved during a potentially power-hungry task, an idea that is similar to being warned to plug in your computer before performing regular updates.

3.5 Biometrics Lifecycle

The biometrics used in the Transparent Authentication Framework progress through a series of stages in the biometric lifecycle, as pictured in Figure 3.7.

Each stage determines how a biometric sample is used and for what purpose. In some cases, the same biometric sample may not be used more than once. For instance, those samples provided in Enrolment to train the classifiers may not be used to adjust device confidence during the Continuous, Transparent Authentication stage. Each biometric used may spend a different length of time in the initial stages, depending on device use. Each biometric is ready for use once sufficient samples have been gathered to contribute to device confidence calculation.

3.5.1 Enrolment

The Enrolment stage marks the beginning of the biometric lifecycle. In this stage, the device owner provides samples of the biometric patterns used in the Framework that, once provided, are then used to train the pattern classifiers. Initially, authentication and task access during Enrolment is dependent on an explicit authentication means, such as a challenge question or password, since the device confidence has not yet been calculated.

The Enrolment stage is also used to educate the device owner on the transparent authentication method, including what biometrics are being gathered, how to determine current device confidence and the purpose of the explicit authentication feature. It is at this point that the device owner should also be warned that while device confidence is being increased (i.e., the device is "getting to know" its owner), explicit authentication will happen more frequently, but will reduce over time. This stage should only be entered in the early stages of using the biometric in the Framework. Once the initial biometric patterns have been gathered, the classifier training stage begins.

3.5.2 Bootstrapping

During bootstrapping, biometric samples are gathered from the user and classified. This continues until sufficient data is available for the biometric to be used in the overall device confidence calculation. It is distinct from Enrolment because it requires a trained classifier.

This stage is when newly gathered biometrics resulting from regular device use are compared to those provided during Enrolment, and the device confidence is increasing to a point where regular device tasks are allowed. Access to applications, tasks, and data is restricted in this stage, but can be accessed via the explicit authentication method. Each time a viable biometric pattern is seen, it is used immediately to increase (or decrease) device confidence. This continues until the minimum confidence threshold is reached, as set by the device owner during Enrolment.

This stage holds a certain amount of security risk since there is not yet a device confidence to use to allow or disallow tasks, which implies that all tasks are available to all users. However, the device will have minimal functionality since device confidence defaults to 0% in new implementations, so the risk is minimized. The other concern is owner frustration – it is likely that the device owner will be required to use the explicit authentication method more frequently in this stage, which may lead to frustration and concern that transparent authentication is no less frustrating than password and PIN entry.

3.5.3 Continuous, Transparent Authentication

This represents the normal working of the continuous, transparent authentication system, in which new biometric patterns gathered as the device is used are presented to the trained classifier model. Prior to the start of this stage, the device confidence has been increased to the user–selected minimum confidence threshold via the Bootstrapping phase. The pattern classification results are used to update the device confidence, and then to allow or disallow access to tasks and data on the device.

3.6 Design Considerations

This section describes options for the biometrics and pattern classifiers that may be used in the Framework.

3.6.1 Biometrics

The Framework uses one or more biometrics to make decisions regarding the identity of the current mobile device user. The Transparent Authentication Framework supports the use of more than one type of biometric in order to reduce the error rates seen with many behavioral biometrics. The type and number of biometrics is not specified in the Framework; they should be chosen on an individual basis based on the needs of the specific authentication system being developed. The biometrics are interchangeable so that future improvements to existing or discoveries of new biometrics may be leveraged to reduce the error rates and improve security levels beyond those provided by passwords and PINs. While the number and type of biometrics is meant to be flexible, the biometrics used in the Framework should have a majority of the following properties:

Universality: each mobile device owner should have the characteristic being measured;

Distinctiveness: any two people should have sufficiently different characteristics that their identity can be *verified* (as opposed to matching an identity to the person using the device);

Collectability: the characteristic can be measured in a quantitative manner;

Acceptability: device owners should consent to using the characteristic to identify them on their device;

Transparency: the characteristic can be collected while the device owner goes about other tasks;

Minimality: collecting the characteristic should not require additional hardware other than that provided by the mobile device alone.

The first four properties are common to all biometrics [59]; the last two are specific to this Framework.

Possibilities for the biometrics chosen for this Framework include, but are not limited to, device use, keystroke dynamics, gait analysis, and voice–related patterns. These are all behavioral biometrics since most physiological biometrics require explicit participation, such as providing a fingerprint or iris for scanning, adopting a specific facial expression, orientation, or lighting condition for facial recognition. Furthermore, physiological biometrics often require an enrolment phase that has its own security concerns such as confirming that the person providing the biometric is the genuine owner of the device. Therefore, most physiological biometrics conform to the first four properties listed above, but fall short on the last two: transparency and minimality. Since these are the two that are most related to

a continuous, transparent authentication method, physiological biometrics were disqualified from use with this Framework. Should these biometrics adhere to the need for transparency and minimality in the future, they may be added to this Framework without changing its structure.

While the suggested behavioral biometrics all have the required properties, two stand out as ideal for use with mobile devices: keystroke dynamics and speaker verification. Both take advantage of common mobile device tasks: typing and speaking. Keystroke dynamics can be gathered while the device owner types emails and text messages, and voice patterns may be gathered during phone calls, recording voice memos, or using voice–activated search capabilities such as those provided by Apple's Siri[1] and the Google voice search application[2]. Neither biometric requires anything beyond what is already provided on the mobile device, specifically a keyboard and a microphone, thereby fulfilling the minimality requirement. The outstanding properties are collectability, acceptability and distinctiveness; Chapters 4 and 5 in this dissertation have provided initial evidence to conclude that keystroke dynamics and speaker verification also have these properties.

Keystroke dynamics is straightforward since it can be gathered while the owner goes about common tasks. Voice patterns, however, are somewhat more complex because the biometric itself is more complex. Work on speaker verification has the most parallels to a transparent authentication mechanism for mobile devices because it is concerned only with *verifying* the speaker's identity rather than knowing their identity (speaker recognition) or discovering their role in a conversation (conversation analysis). Thus, speaker verification along with keystroke dynamics are recommended as the biometrics for the Transparent Authentication Framework, although it is the intent that these could be replaced with other biometrics as required. Furthermore, it is suggested that whatever biometrics are chosen for use be combined into multimodal biometrics since this is a widely accepted way of reducing error rates seen with individual biometrics.

3.6.2 Pattern Classifiers

Classifier algorithm choice is non–trivial and depends on the biometric chosen and the distinctiveness of the feature vectors calculated from them. Feasibility studies into the applicability of various classification algorithms should be performed after the biometrics used have been chosen. Important considerations for initial classifier choice are speed of classification, ease of development, processor and memory requirements, and accuracy. Suggested initial classifiers to test are k-nearest neighbor, decision trees, and Naïve Bayes variations since they meet these requirements. Justification for these choices is given in Chapters 4

[1] http://www.apple.com/ios/siri/
[2] http://www.google.co.uk/intl/en_uk/mobile/voice-search/

and 5, that describe classifier comparisons for keystroke dynamics and speaker verification on iPhone and iPod Touch devices. Neural networks may be investigated, but are likely too computationally intensive for use on a mobile device. More information on classifier selection properties can be found in Chapter 2.

3.7 Summary

This chapter has presented the Transparent Authentication Framework, a device and operating system independent model that allows developers to create a continuous, transparent authentication mechanism that fits the needs of their particular implementation. The Framework was described in terms of data structures and their associated processes. The concept of *device confidence* was presented, and a method for mapping device confidence to particular tasks on the device in question was offered. The biometric lifecycle was described, with particular focus on how a user might work with an application that was based on the Framework. Specific mention was made of the design considerations specific to the Bootstrapping stage of the lifecycle, since this stage may require more attention to user education in order to avoid misconceptions regarding the frequency of explicit authentication method use.

Chapter 4

Keystroke Dynamics Feasibility Study

This chapter provides details on the study performed to determine whether keystroke dynamics is a suitable behavioral biometric for use in the Transparent Authentication Framework. First, the Keystroke Dynamics Feasibility study[1] is introduced, then details of the study's methodology are given including participants, apparatus, materials, and procedure. The study's results are then presented, along with a discussion of their applicability. Finally, the results of the keystroke dynamics feasibility study are related to the Transparent Authentication Framework and to this research as a whole.

4.1 Study Goals

This study was designed to answer two research questions:

1. Is there sufficient distinctive information in the keystroke dynamics of the study participants to justify using it as a biometric in the Transparent Authentication Framework?

2. Can an "optimal" classifier be identified for the keystroke dynamics behavioral biometric? "Optimal" is defined as the classifier with the lowest error rate that also respects the mobile device environment's limitations in processor speed and available memory.

The answer to the first research question will provide a partial answer to the first overarching hypothesis for this dissertation, H1, as identified in Section 1.4.1. As such, this study was used to determine the feasibility of keystroke dynamics as an authenticator on the iPhone and iPod Touch prior to developing a full implementation. The answer to the second question

[1] University of Glasgow ethics approval number FIMS00760.

above provides a method for evaluating the feasibility by identifying which classifier may be best deployed in implementing the Transparent Authentication Framework, within the limits of a particular device and operating system.

Answers to both of these questions required access to a corpus of keystroke dynamics measurements from a soft keyboard on a mobile device. Such a corpus does not yet exist for research use; thus, another goal of this study was to create a corpus of keystroke dynamics information from typists using soft keyboards. The results of this study have been used to tailor the feature vector and pattern classifier for the keystroke dynamics biometric used in the Transparent Authentication Framework.

4.2 Study Design

The study has two parts: the first part gathered the required keystroke corpus; to this end an iPhone application was created that allows the user to type text into a textbox using the standard iOS soft keyboard. The second part of the study presented the gathered patterns to five pattern classifiers in order to determine whether there was distinctive information in the keystroke patterns of mobile device typists. The latter part of the study provided information used to answer the two research questions that drive this research.

The data gathering part of this study used a between–groups experimental design [152, p. 74] in which each participant was in turn considered the *owner* of their device and the remaining participants made up the group *rest-of-world*. Each of the participants used either an iPhone or an iPod Touch during the experiment. Both iPod Touch and iPhones were included in the study because they use the same type of keyboard and operating system, and are functionally equivalent for the purposes of this study. All devices had a single user, and used the standard soft keyboard provided by Apple. The data gathering exercise resulted in eight datasets, each of which contained an owner and rest–of–world class. The gathered datasets were then presented to the pattern classifiers chosen as part of the second part of the study. The independent variable for the study was the classifier used on each dataset, which depended in turn on which participant was considered the owner. The dependent variables were the classifier error rates, which depended on the participants' typing metrics.

4.2.1 Participants

The first part of the keystroke pattern gathering project involved eight participants. The five female and three male participants ranged in age from mid-20s to late 50s, and also ranged in experience in typing on the iPhone or iPod Touch from beginner (had never used one before this experiment) to expert (repeated daily use). All participants were native English

speakers, and were instructed to type in English during the study. The participants were volunteers who provided their own mobile devices and who were not paid in any way for their participation.

4.2.2 Apparatus and Materials

There were four iPhone 3GSes and four iPod Touch 3rd Generations with a minimum of iOS 3.0 used in this study. The devices used the standard iPhone and iPod keyboards without modification of any kind. The pattern classification part of the study used MatLab release R2012b and the standard pattern classification algorithms found in the Statistics Toolbox add-on. The pattern classification exercise was performed offline (i.e., not on the mobile device itself) since this was a feasibility study to select an algorithm and determine biometric fitness, rather than an assessment of whether these algorithms are viable on the device itself.

4.2.3 Procedure

Part 1: Data Gathering

The metrics used for this work are *key hold time* and *inter-key latency*, as shown in Figure 4.1. These measures represent the dependent variables for this part of the study. Key hold time is used with keystrokes only; it is a measure of how long, in seconds, the owner tapped a particular key. Inter-key latency is a measure of the time that passes between the release of the first key and the tap of the second key. The key hold time for the keystroke key_1 is calculated by subtracting the time the key is tapped from the time the key is released, as shown in Equation 4.1. The key hold time for the keystroke key_2 is calculated similarly. Inter-key latency is used with bigrams, which are sequences of two characters. This metric measures how much time passes between releasing the first key in the bigram and tapping the second key, and is calculated as shown in Equation 4.2.

$$keyHoldTime_{key1} = key1_{release} - key1_{tap} \tag{4.1}$$

$$interKeyLatency_{key1key2} = key2_{tap} - key1_{release} \tag{4.2}$$

After each participant signed the ethics Information and Consent sheets, the custom-designed application was loaded onto their device and the participant was instructed to use the application as many times per day as they wished to enter keystroke data. Participants were asked

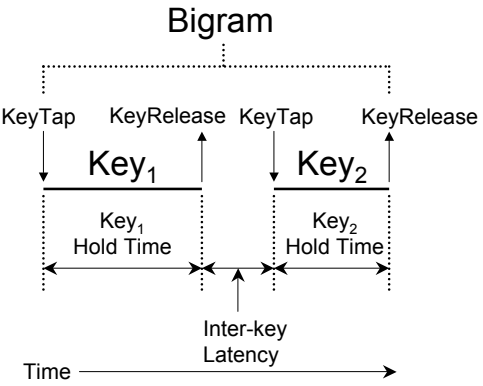


Figure 4.1: Keystroke metrics. Key hold time applies to keystrokes and inter-key latency applies to bigrams.

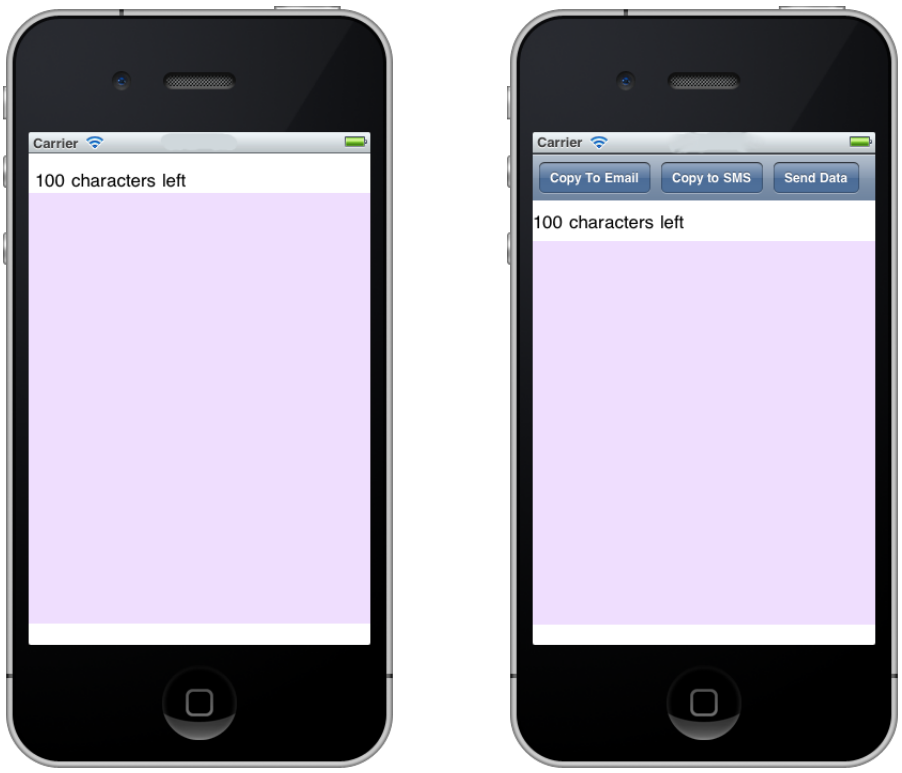
to enter at least 100 characters each time they used the application in order to ensure that sufficient data was gathered to be distinctive. The 100 character minimum was not rigidly enforced and participants could type more or fewer characters if they wished. Screenshots for the two versions of the KeystrokeData application can be seen in Figure 4.2a. The difference between the two application versions was that the second version allowed participants to send their data to the experimenter via the application. In the first version, the data had to be gathered manually, which reduced the possible pool of participants to those physically near the experimenter. Allowing for remote data gathering solved this issue.

The user interface consists of a single text box, as seen in Figure 4.2a, and a counter that represents the number of keystrokes left to type to reach 100 characters. To enter text, the user tapped on the text box and the keyboard appeared. When the participant finished typing, they closed the application by pressing the Home key on the device. After three weeks the data from each device was gathered manually and the application was removed from each participant's device.

The second iteration of the application, as seen in Figure 4.2b, added the ability to copy typed text to an email or text message. The buttons used to implement this functionality are labeled "Copy to Email" and "Copy to SMS", respectively. The third button, labeled "Send Data" allowed the participant to email the SQLite data store containing the keystroke information to the experimenter.

Other than the ease-of-use additions mentioned above, the inner workings of the second iteration of this study were identical to the first iteration. After approximately three weeks, an email was sent to the two study participants asking them to email the SQLite data store to the experimenter using the button on the application. The participants were also given instructions on how to remove the application from their device. All data stores were sent successfully.

The keystrokes used in the study were those available on the standard iPhone and iPod Touch



(a) First Iteration (b) Second Iteration

Figure 4.2: Screenshots of the first and second KeystrokeData application.

soft keyboard, as seen in Figure 4.3. The predictive text function was disabled as part of this study because it skewed the user's normal typing pattern. When predictive text changes the series of characters actually typed by the user, the resulting changed string is replaced as one chunk of text. The result is that the application gathers the timing information for the original string of characters, then gets another string of characters from the predictive text that all have the same key up and key down times, which means that the resulting key hold time is zero. The net effect of predictive text is that the timing information needed for the keystroke pattern is effectively lost. Future iterations of the keystroke gathering application must manage this problem since asking users to disable predictive text is not likely to be feasible.

Another notable change in regular typing functionality seen in the KeystrokeData application is that automatic capitalization was disabled. In normal typing patterns, in order to type an uppercase letter, the user must first tap the Shift key and then the letter they wish to type. This means that the inter-key latency for a bigram containing a uppercase letter would be different than the inter-key latency for the bigram containing the same lowercase letters. For instance, if the user typed the bigram *em*, we would expect that the inter-key latency would be shorter for this bigram than for the bigram *eM* since the user would have to press the

Shift key before the *M* key in the second case. While it is desirable to allow device owners to enable predictive text and auto-capitalization if they wish, the limitations imposed by disabling them are acceptable for a proof-of-concept such as this experiment.

(a) Letters (b) Numbers (c) Symbols

Figure 4.3: The characters available on a standard iOS soft keyboard. The keys outlined in red are not considered keystrokes by the keystroke gathering applications. The English keyboard is shown since the participants were instructed to type in English.

The shift key is not considered a keystroke in this study. The return, space, and backspace keys were all considered keystrokes, and were also allowable parts of a bigram. The space character was included because it is an allowable and frequent character in English. Since the KeystrokeData application did not allow the use of predictive text, participants often corrected their typing errors by using the backspace key. This has the possibility of containing valuable information on the user's typing habits, and thus was considered a particularly rich source of distinctive information about the participant, so it was included as a keystroke. The return character was included as a keystroke for completeness, since it is a valid character on a soft keyboard.

Part 2: Pattern Classification

Once the data had been gathered from the study participants, it was presented to five pattern classifiers in order to determine firstly whether there was enough distinguishing information in the data to justify using keystroke dynamics in the Transparent Authentication Framework, and secondly whether any of the classifiers could be considered optimal for the type of data gathered. The five pattern classification algorithms used in this study were described in detail in Chapter 2. In summary, the five classifiers are Naïve Bayes with kernel density and Gaussian estimation techniques (NB (KD) and NB (Gau), respectively), Decision Tree (DT), and k–Nearest Neighbor with both Manhattan and Euclidean distance measures (k-NN (Man) and k-NN (Eucl), respectively). Each classifier was trained using supervised learning methods, which means that the classifier was trained on a combination of patterns from both the owner and rest-of-world groups and the known classes of each sample were provided during training.

The performance of each classifier was estimated using 10-fold cross-validation methods [153, 154]. These methods can be used to assess whether the results on a particular dataset will generalize to an independent dataset, which is very important due to the study's relatively small dataset. Cross-validation is performed by repeatedly partitioning a dataset into training and testing sets (10-fold implies this is done 10 times), and then running the classifier on each of the 10 data partitions. The mean of the results of each run of the classifier was calculated to produce a final result, which reduces the classifier output's dependence on a particular dataset. The classification problem is two-class; either the pattern belongs to the owner of the mobile device, or it belongs to the rest-of-world.

4.3 Data Acquisition

The keystroke dynamics module was written in Objective-C and is specific to the Apple iPod Touch and iPhone. The gathered data was stored on the device in an SQLite database. Three object types were used to organize and store the keystroke information: Keystroke objects, Bigram objects, and Pattern objects. These objects are related as shown in Figure 4.4. Each of these objects is created as a Managed Object in Objective-C, which gives the developer control over the object's attributes, methods, and relationships. The iOS CoreData framework can be used to save the Managed Objects and their relationships to a simple datastore. This framework automates much of the object management and saving functionality, which frees the developer from managing these often tedious and error-prone tasks. The data is stored on the device in an SQLite database, which can be retrieved from the device and accessed with standard SQLite database management tools.

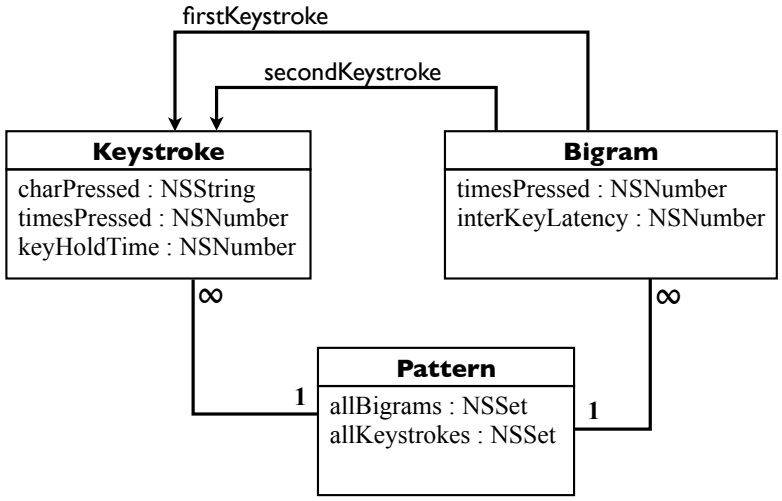


Figure 4.4: Details of Pattern, Keystroke, and Bigram classes.

The Pattern Managed Object (MO) is used to store the information on a series of keystrokes

as typed by the user. The Pattern MO contains references to Keystroke and Bigram MOs, which are used to store the information about keystrokes and bigrams, respectively. A Bigram MO contains references to the two Keystroke MOs that form the bigram. Figure 4.5 shows the relationship between the three object types using the word *science* as an example. Assuming that no other characters have been typed previously, when the user types the first character, *s*, a new Keystroke object is created to represent *s*. Next, the user types *c* and the program checks to see if it already has a Keystroke object representing *c*; since it does not, this object is created. Now that two characters have been typed, the program checks to see if an object representing the bigram *sc* exists – again, it does not so this Bigram object is created by storing references to the *s* and *c* Keystroke objects. As the user continues to type, Keystroke objects for *i*, *e*, and *n* are created, as well as the corresponding Bigram objects (with references to each required Keystroke object) for bigrams *ci*, *ie*, and *en*.

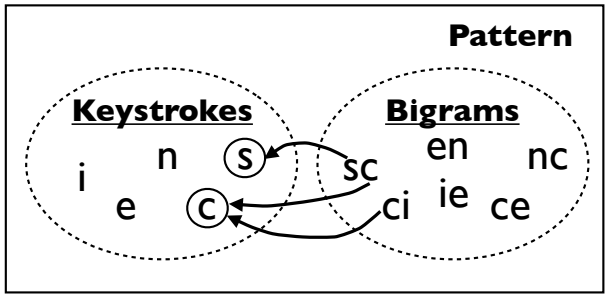


Figure 4.5: Relationship between Pattern, Keystroke and Bigram objects for the word *science*.

Next, the user types *c* again, which is already represented by a Keystroke object. Rather than create a duplicate object, the program recognizes that this Keystroke object already exists and checks to see if the corresponding bigram *nc* exists – this bigram is made up of the previous character typed (*n*) and the current character (*c*). Since the bigram does not already exist, the new *nc* Bigram object is created that contains references to the new *n* Keystroke object and the existing *c* Keystroke object. Keystroke object duplicates are undesirable for two reasons: firstly, because memory is a constrained resource on mobile devices so duplication of data should be avoided wherever possible; secondly, the metric of interest for a single keystroke is key hold time, which is a measure in seconds of how long the user pressed the typed key. With repeated presses of the same key, the metric of interest is the *arithmetic mean* of the individual key hold times for each keypress. The combination of hold times gives a more effective metric for future pattern comparisons because it allows for small variations in the user's typing patterns while minimizing the effect of large changes. Therefore, representing subsequent occurrences of the same character with the same Keystroke object is not only valuable in terms of saving memory, it is also a way of storing the more valuable metric of mean key hold time. The user then types the final character, *e*, and a similar operation as

above takes place: since *e* is already represented, its key hold time is averaged and the new Bigram object for *ce* is created.

At this point, the user has finished typing, and the Keystroke and Bigram objects are bound to a single Pattern object that contains references to each of the Keystroke and Bigram objects created. The user could have continued typing and further Keystroke and Bigram objects would be created as described above; a new Pattern object is created when the user ends the application in which they were typing. Pattern objects are used as a separator between distinct typing instances and to facilitate the creation of a series of typing patterns rather than a single pattern that is used for later comparison to a gathered pattern. This allows for several comparisons to be made in situations where comparison to a single pattern may not provide sufficient confidence to confirm the user's identity.

Keystroke, Pattern, and Bigram objects each store information about their respective contents, as can be seen in the class diagram in Figure 4.4. Keystroke objects are made up of the character typed, the key hold time, and a counter that represents the number of times this character was typed in the current Pattern. Bigram objects contain references to the two characters typed in the order they were typed (this is important since the bigram *th* is not the same as the bigram *ht*), the inter-key latency, and a counter representing the number of times this bigram was typed in the current Pattern. Pattern objects contain two Set objects: one containing references to each of the Keystroke objects related to this pattern, and one containing references to each of the Bigram objects. Optionally, a Pattern object could contain a timestamp representing the time the pattern was gathered, as well as which application was being used at the time the pattern was gathered. Currently, however, this information is not gathered because it is not relevant to the pattern classification task.

Each Pattern object and its related Keystroke and Bigram objects are stored in an SQLite database on the user's iPhone or iPod Touch. This storage method is handled through iOS's CoreData framework. Using this framework ensures that the database structure is compact, and that data is stored in a way that minimizes memory and processor overhead[2].

The relationship between Keystroke, Bigram, and Pattern objects, as seen in Figure 4.4, was designed so that the amount of data stored on the device was minimized. Keystroke objects contain the character typed, the key hold time, and a counter that denotes how many times the key in question has been pressed. As a key is tapped, if that key has not been tapped previously within the session in question, an array is created containing the KeyUp and KeyDown times, the character tapped, and the key counter is set to 1. The KeyUp and KeyDown times are used to calculate the key hold time, as explained in Section 4.2.3, which is then stored in the array. As more keys are tapped, if the keystroke has been seen before,

[2] `http://developer.apple.com/library/mac/#documentation/cocoa/Conceptual/CoreData/Articles/cdTechnologyOverview.html#//apple_ref/doc/uid/TP40009296-SW1`

the existing array is found, and the mean of the existing and new key hold times is calculated. The mean key hold time then replaces the existing hold time in the array. Once the session has finished, new Keystroke managed objects are created for each keystroke array stored. The Keystroke managed object stores the character tapped, the key hold time (which is the mean in the case of multiple taps of the same key), and the number of times that the character has been tapped. Note that the KeyUp and KeyDown times are *not* stored in the Keystroke objects – these times are discarded since they are only needed to calculate the key hold time, which is performed as the key is pressed. Furthermore, storing these times would allow the characters to be sorted in chronological order according to when each one was tapped, which means that the original message as typed by the user could be retrieved. By storing only the length of time each key is tapped, the user's privacy is protected to a greater extent since it would be more difficult to sort the characters into their original order.

Once the data was gathered from the users in both study iterations, it was pre-processed offline on an iMac with a 2.66 GHz Intel Core 2 Duo processor, 4 GB of RAM, running Mac OS X version 10.6.8. Pre-processing began with extracting all relevant timings and keystroke or bigram frequency counts from the objects stored in the SQLite database for each user. Then, each study participant in turn was designated the "owner" and their five most frequently typed keystrokes and 10 most frequently typed bigrams were selected; the other seven participants' data was considered part of the "rest-of-world" dataset. This information was used during feature extraction to determine which timings would be used for that owner.

Typing mistakes were not filtered out during the pre-processing stage since dynamic text analysis does not, by definition, provide a fixed text string to type in order to provide a comparison template for mistakes. The use of the backspace key could be used as an indication of a mistake, but instead it was used as a keystroke and treated no differently than any other keystroke. The purpose of the study was to gather the owner's raw typing pattern. If they commonly make a mistake such as typing *teh* instead of *the*, then this is a relevant part of the owner's typing pattern that should not be lost to auto-correct or predictive text changes. Similarly, uppercase letters are treated as a different keystroke from their lowercase version, so use of auto-capitalization would replace a lowercase keystroke with an uppercase keystroke in some instances, thereby skewing the user's actual letter frequency counts.

Once the data was organized into the owner and rest-of-world datasets, they were examined for outliers. An outlier was defined as any timing value greater than the keystroke or bigram mean timing plus three standard deviations (SD). In addition, a value was considered an outlier if it was greater than 0.5 seconds for keystroke timings and greater than 3.0 seconds for bigram timings. The reason for the latter inclusion in the definition of an outlier is because the data represented two distinct user behaviors: typing and not-typing. Typing behavior is seen when small timing values near the mean are seen; non-typing behavior is seen when larger values are seen, which indicates a pause in the user's typing. These large timing

values may be because the user was distracted while typing, perhaps by an interruption, or to pause to gather his or her thoughts, or by a telephone call. By removing the outliers, we represent the typing behavior and remove the non-typing behavior and thus our data represents the behavior we are interested in using. The reason behind the specific cutoff time of 0.5 seconds for keystrokes was because holding a key longer results in a repeated key on the iOS soft keyboard. The reason for selecting 3.0 seconds for the bigram cutoff time was because this is thought to represent a distraction, where one or two seconds may just be a slower typing speed. In both cases, those values greater than the mean plus 3 SD are non-representative of the user's typing pattern.

Feature extraction is a data processing step that converts the raw keystroke and bigram timings into a feature vector that is a compact representation of the data. The feature extraction step created the feature vectors as follows: first, one user was designated the owner of the device and their five most frequently typed keystroke and 10 most frequently typed bigram timings were concatenated into the feature vector. Equation 4.3 shows the structure of the feature vector, where k_i is the key hold time for key i and b_j is the inter-key latency for bigram j. The data set from each user was then processed to extract only the timings for the *owner's* features, since these features are the only ones that would be available on the owner's device. Other options for the 15 features include the five most frequently typed keystrokes and 10 most frequently typed bigrams for the English language, although a simple comparison test on the device owners showed that better results were seen with the owner's most frequently typed keystrokes and bigrams, as evidenced by lower classification error rates with these latter features. Another option for feature vectors was to determine the typist's most frequently typed 15 features while they type, but this is not only computationally expensive, but does not allow for easy comparison to existing feature vectors that do not have the same keystrokes and bigrams represented. The owner's most frequently typed keystrokes and bigrams were used because this is the only information that will be available on the device; it is unlikely that a representative sample of rest-of-world typing patterns could be made available for comparison purposes. Additionally, a change in these most frequently typed typing patterns could represent another user typing on that device, although this conjecture's proof is left for future work.

The feature vector contains five keystrokes and 10 bigrams for several reasons. First, the length was limited to 15 timings to keep the vector manageable for on-device storage during pattern gathering, and to allow for faster processing. The choice to use five keystrokes and 10 bigrams was made because there are only 62 characters that can be typed on an iPhone soft keyboard, where there are $62 * 62 = 3844$ possible bigrams, which gives a larger set from which to select features. Various combinations of 15 features were considered, from zero keystrokes and 15 bigrams up to 15 keystrokes and zero bigrams, but there were few if any gains in error rates with this manipulation. The exception is that the results of this

study show that key hold time is not a distinctive characteristic on the iPhone soft keyboard, and thus using 15 bigrams and zero keystrokes may improve results, although proving this is outside the scope of this work.

$$f_{user} = (k_1, k_2, \ldots, k_5, b_1, b_2, \ldots, b_{10}) \tag{4.3}$$

The feature extraction step was repeated eight times in total – once for each study participant, who is considered the owner for that data run. The final result was eight distinct groupings of the entire dataset, each of which contains an owner dataset and a rest-of-world dataset.

4.4 Results and Analysis

This section contains the results of this study and the analysis of these results in terms of the Transparent Authentication Framework. It begins with a discussion of the pattern classifiers tested, then presents the reported error rates for each pattern classifier. The data itself is discussed initially to provide context for the statistical and error rate analyses that follow. These error rates are discussed as values in themselves along with their meaning, and are also tested to determine whether selection of a specific classifier by error rates is also statistically significant.

In total, 251 keystroke patterns were gathered from eight participants; Table 4.1 shows the number of keystrokes, bigrams, and patterns gathered on each device. The largest amount of data at 487 KB was gathered from iPhone3, with a total of 61 patterns containing 10233 keystrokes and 8378 bigrams. This represents a mean pattern size of approximately 167 characters – well above the desired 100 characters per pattern. The smallest data store at 135 KB contained 14 patterns that referenced 2182 keystrokes and 1833 bigrams. The mean size of each pattern for this device is 156 characters, which is still well above the desired minimum. The participant who used iPhone4 had little experience using a soft keyboard prior to this study. The data provided by this participant is encouraging; it shows that even users who do not type very much can still be a rich source of keystroke information.

One of the main goals when designing both the KeystrokeData and full keystroke dynamics applications was to minimize the amount of data stored in order to minimize the memory and processor loads on the device. Using Objective-C's CoreData helps create small object footprints, but the sheer volume of possible data gathered on a frequently used device may be a limiting factor. To this end, the size of each SQLite data store was noted at the end of each iteration of the study. As can be seen in Table 4.1, the largest data store was 487 KB and contained a considerable amount of data: 61 patterns that held 10233 keystrokes and 8378 bigrams.

Device	Keystroke Count	Bigram Count	Pattern Count	Data Store Size (KB)
iPhone1	6304	5410	36	307 KB
iPhone2	5863	5737	41	352 KB
iPhone3	10233	8378	61	487 KB
iPhone4	2182	1833	14	135 KB
iPod1	3785	3706	19	180 KB
iPod2	6789	6409	32	315 KB
iPod3	3191	2846	27	201 KB
iPod4	4944	4207	21	225 KB
Totals	**43291**	**38526**	**251**	**2202 KB**

Table 4.1: Number of Keystrokes, Bigrams and Patterns collected.

Table 4.1 shows that the largest datastore also contains the most patterns, keystrokes and bigrams, as is expected. The concern is that maintaining links to typing patterns may increase the amount of on-device memory used, which may negatively affect the user's experience with an application based on the Transparent Authentication Framework. The small data sizes for the amount of data gathered show that this is not likely to be an issue given that the iPhones and iPod Touch devices available today contain either 16, 32 or 64 GB of available memory. In order to further minimize the memory load on the device, the keystroke gathering application should keep the minimum amount of information needed to authenticate the device user.

Classifier Performance Metrics

The classifier performance metrics used for this study were False Accept Rate (FAR), False Reject Rate (FRR), Equal Error Rate (EER), and Area Under the Receiver Operating Characteristic Curve (AUC), as described in Chapter 2. FAR and FRR are known to be overly sensitive to unbalanced datasets where the number of positive and negative patterns is unequal [134]. This sensitivity results in a over- or under-estimation of errors in cases where the dataset balance is unequal. In the case of the keystroke data gathered for this study, the dataset balance is highly skewed to negative samples (i.e., those samples not belonging to the device owner) as shown in Figure 4.6. This reflects the case when the entire population is considered since there will always be only one device owner, but many, many more patterns in the possible rest-of-world class when all other people are considered part of rest-of-world. The result is that the classifiers may have over-estimated false rejects and underestimated false accepts. EER is also sensitive to unbalanced datasets since it is related strongly to FAR and FRR. AUC, however, is not as affected by unbalanced datasets because the two values upon which it is based (FAR and True Accept Rate) are equally affected by the dataset dif-

ferences [134]. Furthermore, AUC is considered discriminatory and has been suggested as a better measure of classifier performance [137, 155]. Therefore, while all four performance metrics are reported, the AUC values are considered the more discriminatory and accurate measurements for this study.

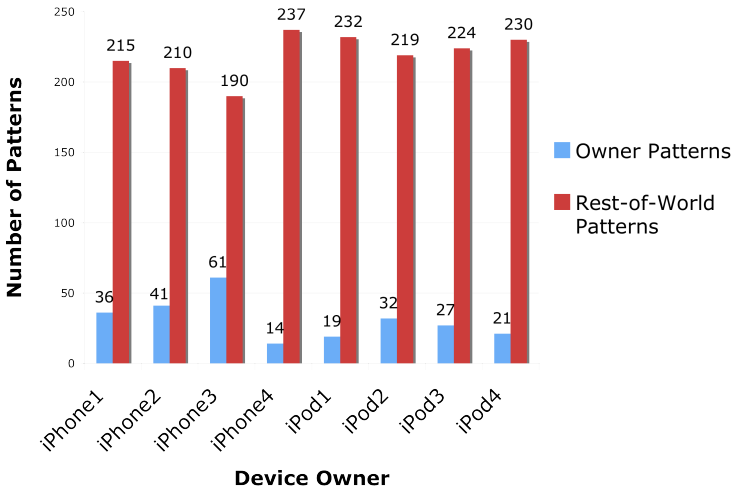


Figure 4.6: Proportion of owner and rest-of-world patterns for each device owner. There is a much larger number of negative (rest-of-world) examples than positive (owner) examples in the dataset, thus skewing FAR, FRR and EER values.

Table 4.2 shows the results when the owner and world datasets were presented to the pattern classifiers discussed in Chapter 2. Each row grouping represents a separate device owner, and the results in that row of the table are from classifying the subsets of owner and world data organized according to that owner's five most frequently typed keystrokes and 10 most frequently typed bigrams. The results in the table are the median values for 10-fold cross-validation, which takes the entire dataset for that owner and divides it into 10 randomly chosen but equal-sized subgroups. Each subgroup is then divided – one-third is reserved for training and two-thirds is reserved for testing. Each classifier, in turn, was then trained on the training set and tested on the testing set and the EER, FAR, FRR, and AUC values were calculated from the results.

Related research in the field of keystroke dynamics on mobile devices has reported FAR and FRR values between 0% and 2.5%, and EER values between 9% and 24% [18, 82, 104–106, 113]. Many of the EER and FAR values reported in Table 4.2 fall within these ranges, but the FRR values are higher, which can be explained by the inbalance in the dataset, as described previously. Few studies report AUC, which is considered more distinctive for this research, therefore a comparison of AUC ranges to related work is not possible.

Due to the use of the owner's most frequent keystrokes and bigrams as features, there may be cases where a non-owner did not type those particular keystrokes or bigrams. In this case, NaN (Not a Number) values were used in places where there was no data (i.e., the typist did

Result	Owner	Classifier				
		Naïve Bayes		Decision	5–NN	
		Gaussian	Kernel Density	Tree	Euclidean	Manhattan
FAR (%)	iPhone1	26.06	11.34	3.55	0.00	0.00
	iPhone2	13.04	5.08	5.08	1.45	1.45
	iPhone3	18.55	9.68	16.13	1.61	1.61
	iPhone4	0.00	0.00	0.64	0.00	0.00
	iPod1	7.23	4.61	2.63	0.66	1.32
	iPod2	17.36	7.64	8.33	2.78	2.78
	iPod3	2.70	1.37	1.37	0.00	0.00
	iPod4	29.61	7.89	5.26	0.00	0.00
FRR (%)	iPhone1	43.56	56.44	87.12	100.0	100.0
	iPhone2	23.08	42.31	42.31	84.62	76.92
	iPhone3	25.00	27.50	25.00	80.00	80.00
	iPhone4	0.00	0.00	37.50	100.0	100.0
	iPod1	50.00	75.00	66.67	100.0	91.66
	iPod2	30.00	40.00	55.00	75.00	70.00
	iPod3	25.00	31.25	35.41	100.0	100.0
	iPod4	33.33	83.33	83.33	100.0	100.0
EER (%)	iPhone1	30.99	29.08	34.22	41.19	38.31
	iPhone2	16.66	16.66	24.35	8.60	5.26
	iPhone3	25.00	20.49	17.97	21.11	10.79
	iPhone4	0.00	0.00	29.31	0.00	0.00
	iPod1	23.03	33.33	40.41	30.34	24.11
	iPod2	25.70	20.00	36.32	30.74	26.31
	iPod3	18.58	17.57	20.11	21.82	14.88
	iPod4	32.24	32.45	41.13	37.50	48.81
AUC (%)	iPhone1	77.20	78.11	68.23	61.31	65.16
	iPhone2	91.30	90.86	79.63	94.87	96.66
	iPhone3	79.96	86.05	84.63	80.63	93.99
	iPhone4	100.0	100.0	79.16	100.0	100.0
	iPod1	86.29	77.74	64.58	79.25	84.77
	iPod2	80.41	84.66	68.44	76.27	81.27
	iPod3	86.40	87.50	87.51	88.20	92.20
	iPod4	71.33	72.91	64.97	67.84	52.44

Table 4.2: Pattern Classifier Results, median values of 10-fold cross-validation. Rest-of-world participants are repeated in both training and testing sets (data is not repeated).

not type the particular keystroke or bigram in question). Two other options for representing this lack of data were tested: using zeros or a very large number in place of no data. Using numeric values to represent no data is problematic because it skews calculations of mean,

median and mode for the data after it has been gathered, and may have a small effect on the error rate calculations. Similar error rate calculations as shown in Table 4.2 were also performed for the zero and large number datasets; the result was slightly better error rates with the NaN dataset, which is why it has been used for the rest of the calculations for this work.

Table 4.2 show that some owners had a very distinct pattern (e.g., iPhone3 with low EER values for both 5–NN classifiers) but others were more similar to the rest–of–world values (e.g., iPhone3 with values near 40% for the 5-NN classifiers). Overall, since these results are averages of many data runs, these results show that keystroke dynamics on a soft keyboard is dependent on the type of classifier, and is unlikely to be certain enough to allow most owners to use it as a *sole* measure of whether the owner is using the device.

The first and second results groupings show the FAR and FRR values for each owner. In general, the FAR values are quite low and the FRR values are quite high. This result is due to the unbalanced dataset, and confirms the assumption in the previous section that the classifiers given unbalanced data will underestimate false accepts and overestimate false rejects. For this reason, the FAR and FRR values will not be considered when selecting an optimal classifier; they are included here for completeness and to show the results of using unbalanced datasets in pattern classification tasks.

In order to allow comparison with other studies, the Equal Error Rate (ERR) for each classifier and owner is also reported in Table 4.2. The EER values reported are relatively high, with many values higher than 40%. The lowest EER values of 5.26% and 8.26% can be seen for iPhone2 as owner with the two 5–NN classifiers (iPhone4 values are not being considered since this participant provided very little data, which skews classifier error rates). The other values reported are significantly higher than is acceptable for a single authentication method, but this is expected since the unbalanced dataset that affects FAR and FRR also affects EER since it is dependent on FAR and FRR. However, the two main goals of this study were to determine which of the pattern classification algorithms works best with the data gathered during this study, and to test whether keystroke dynamics is a viable part of a multimodal authentication system. The EER values, while high in some cases, support using keystroke dynamics as part of a multimodal biometric because they were also quite low in some cases.

The final error rate calculated for the gathered data was AUC, or Area under the Receiver Operating Characteristic Curve. For this error rate, a higher percentage shows that the classifier is more accurate. The percentages shown in the final row grouping in Table 4.2 are wide–ranging, with values as low as 52.44% for iPod4, and as high as 96.66% for iPhone2 both with the 5–NN (Man) classifier, when not considering the iPhone4 results. Some of this wide range of values can be attributed to the small dataset, although the overall results are encouraging. The two 5–NN classifiers performed well for all owners; these are the two

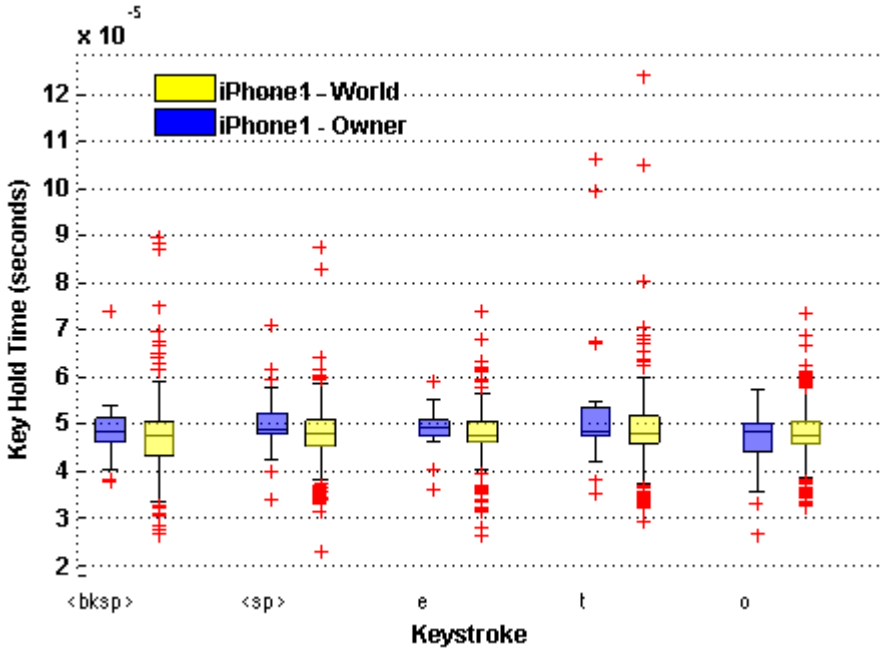
classifiers recommended for the keystroke data.

In some cases the EER and AUC values are relatively high and without much variance between classifiers, such as those values seen for iPhone1 as the owner. The EER values here range between 29.08% and 41.19% and the AUC values range between 61.31% and 78.11%, which means that nearly one-third of this owner's patterns are consistently misclassified no matter what classifier is used. This implies that the owner of iPhone1 does not have a particularly distinctive typing pattern, and that it is therefore difficult to recognize their pattern from those of rest-of-world. This further implies that there are likely some owners for whom keystroke dynamics will not be distinctive enough to allow identification or authentication, possibly even when considered in conjunction with other biometrics. Many biometrics suffer from this difficulty – it is known as *failure to enrol* [59].

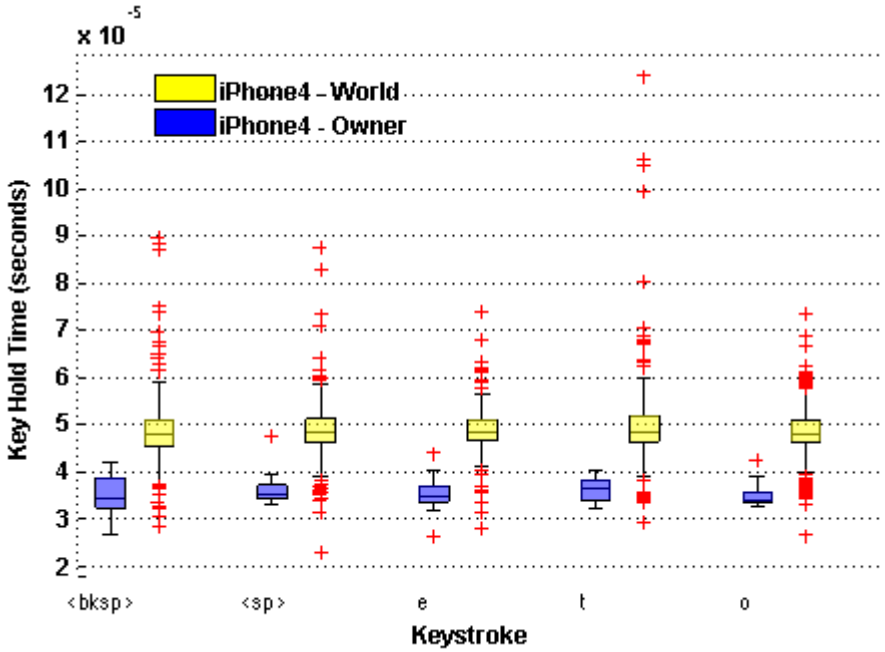
The EER values in Table 4.2 support the choice of either of the 5-NN classifiers as the optimal classifier for the data in this study, since these are the lowest values for most owners. As discussed previously, FAR and FRR as measures of quality (and thus EER since it is dependent on FAR and FRR) have been shown to be sensitive to unequal or skewed datasets in which there is an unequal number of positive and negative patterns presented to the classifier [134]. The sensitivity of EER to unequal representation in datasets means that it must be considered with caution when attempting to determine the optimal classifier from this measure alone. The AUC values, which are less sensitive to unbalanced datasets, support the choice of either 5-NN classifier since they have the highest AUC values for most owners. This discrepancy between classifier choices when considering different error rates is likely to change when a larger study with more participants who provide more data is undertaken. Therefore, the answer to the research question regarding the optimal classifier is k-NN with either Euclidean or Manhattan distance measures.

The boxplots for iPhone1 and iPod2 in Figures 4.7a and 4.8a show mean key hold time and inter-key latency values for owner and rest-of-world patterns that are quite similar. In each case, the median of the means (represented by the horizontal line in each boxplot) for the two groups are quite similar, which implies that it would be harder to distinguish the owner patterns from other patterns during classification. On the other hand, the boxplots in Figures 4.7 and 4.8 show that each owner's mean key hold time and inter-key latency values are distinct from the same values for the same keys for the rest-of-world group. This may lead to higher than acceptable error rates. The distinctive nature of these latter two examples shows that keystroke dynamics is a plausible biometric for authentication since there are cases where patterns are different enough to be separated from others. As such, these results show that further study is worthwhile and likely to deliver a viable tool, which was the purpose of this work.

The results of this study show that there is sufficient data in owner keystroke dynamics on a



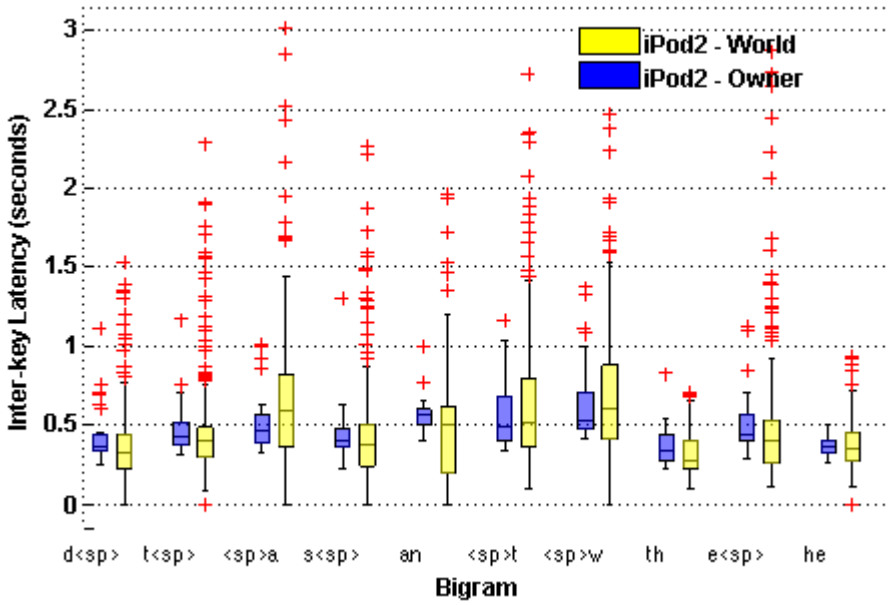
(a) iPhone1



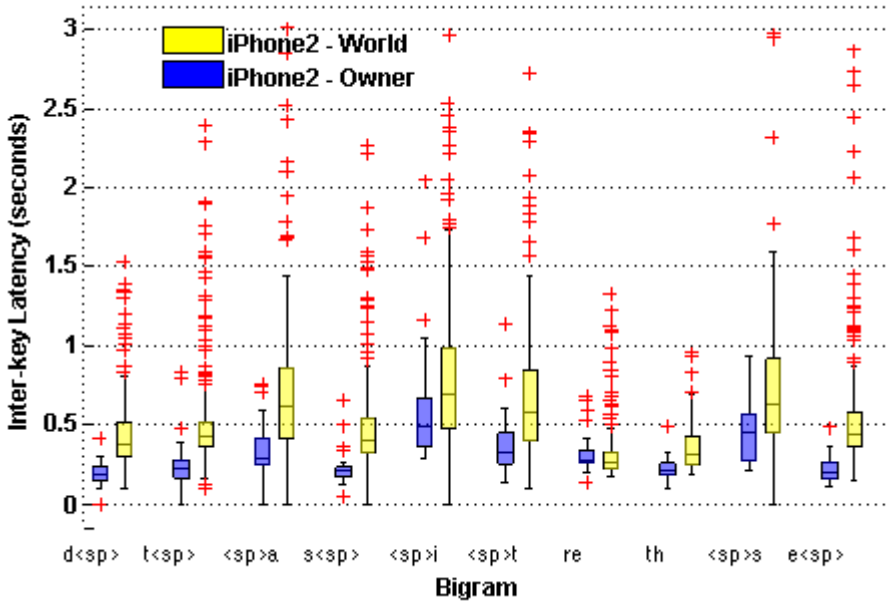
(b) iPhone4

Figure 4.7: Mean key hold times for Owner and World patterns. The keystrokes chosen in each case are based on the five most frequently typed keystrokes for the owner in question; they are different for each chart.

soft keyboard to support further work on using it as an authenticator on a mobile device with a soft keyboard, particularly in conjunction with other biometrics. As with other keystroke



(a) iPod2



(b) iPhone2

Figure 4.8: Mean inter-key latency times for Owner and World patterns. The bigrams chosen in each case are based on the 10 most frequently typed bigrams for the owner in question; they are different for each chart.

dynamics studies, the results do not support using keystroke dynamics as the *sole* authenticator or identity verification tool because, while the difference in patterns between owner

and rest-of-world is promising, there is not enough difference between patterns to identify the device owner with sufficient confidence.

While the majority of the performance metrics shown in Table 4.2 support the conclusions drawn above, the results for iPhone4 as owner are markedly different. The lowest FAR and FRR values were 0.00%. The low error rates are because this owner's typing rates are significantly different than those of the rest-of-world group. Thus, their patterns are easily distinguished from more experienced typists. The result of this is that any keystroke dynamics implementation should adapt to changes in the owner's typing patterns as the owner becomes a more proficient typist, but not so flexible that the error rates increase to unacceptable levels.

Comparison of the bigram mean inter-key latency for the owner and rest-of-world classes in Figure 4.8 shows that there is a difference between the average typing speeds between the groups, with the owner of the device generally being a faster typist for these bigrams. This result can be attributed to two factors: first, the device owner is more familiar with their own device than others may be; second, the use of the owner's most frequently typed bigrams results in faster and thus more distinctive typing speeds since the rest-of-world members may not type those bigrams as often. This may support the creation of another metric for use in future systems: that of keystroke and bigram frequency. If the owner's feature vector consisting of their most frequently typed keystrokes and bigrams were to change suddenly, this may indicate that someone other than the owner is using the device. Such a result could likely be combined with the results of the key hold time and inter-key latency to improve the results of pattern matching, although further study needs to be performed to prove or disprove this conjecture.

Large error bars on the rest-of-world datasets in Figures 4.7 and 4.8 imply that there is variance in the timing values for the rest of the world, which is to be expected since the mean and standard deviations take into account seven different keystroke patterns. However, the smaller error bars seen on the owner datasets in the same figures show that, with careful training, the owner's keystroke pattern can be determined with far fewer possible errors. This is significant because the goal was to separate the owner from the rest of the world, not to identify exactly who created the pattern in question.

The mean timings shown in Figures 4.7 and 4.8 have grouped together the seven rest–of–world members into a single result for clarity, but in doing so we have in essence created an "normative world user" and compared them to the device owner. While the results shown are valid, demonstrating that each owner in turn can be distinguished from each member of rest-of-world would be more realistic, since it is unlikely that an exactly average rest-of-world user will choose to use the device. Instead, comparing the owner to the extreme cases within the rest-of-world class (i.e., those users who were very close and very far away from the owner's mean metrics, in terms of a measure such as Euclidean or Manhattan distance)

would provide a better understanding of the proportion of other users from which the owner can be distinguished.

Statistical Significance

The goal of statistical significance tests is to determine whether the differences seen between two sets of data exist because of the differing data treatments or because of other random effects. Since one of the goals of this study was to choose a optimal pattern classifier for the keystroke data, the comparison for statistical significance is between the AUC and EER for each owner and for each of the pattern classifiers tested. This method follows the classifier comparison methodology described by Killourhy & Maxion in [156].

For the AUC and EER values calculated for this experiment, there are two important features that a statistical significance test must have, as follows:

1. Non-Parametric: the data gathered is not parametric (i.e., does not follow a normal or Gaussian distribution) as evidenced both by the fact that the EER and AUC values range between 0 and 1 and the results of the Kolmogorov-Smirnov test [152, p. 160] as shown for each of the calculations in turn, as seen in Table 4.3.

2. Pairwise Tests: For each owner data, the comparison is between the AUC or EER for each classifier – this means that the results for the optimal classifier (the one with the lowest EER median and highest AUC median) should be compared to the AUC and EER for each of the remaining four classifiers in turn, which implies using a test that will compare pairs of values.

Metric	**Classifier**				
	Naïve Bayes		Decision Tree	5–NN	
	Gaussian	Kernel Density		Euclidean	Manhattan
EER	0.0226	0.0226	0.0053	0.0026	0.0026
AUC	0.0000	0.0000	0.0001	0.0001	0.0002

Table 4.3: Results of the Kolmogorov-Smirnov test of distribution shape for EER and AUC values. Since each value is less than the $\alpha = 0.05$ significance level, then each distribution of data is considered significantly different from a normal distribution (i.e., the distribution for these values is non–parametric).

Based on these two requirements, the Wilcoxon Signed-Rank test has been chosen as the statistical significance test for the keystroke data EER and AUC values. The Wilcoxon test is the non-parametric version of the dependent t-test and is used in cases where there are two conditions (i.e., two different classifiers) and the same participants have been used for each

condition [152]. Non-parametric tests make fewer assumptions about the distribution of the analyzed data, although often at the cost of the descriptive power of the test. The significance level for the Wilcoxon tests was set at $\alpha = 0.05$ as with Killourhy & Maxion's procedure.

The median of the EER and AUC data per classifier in Table 4.2 was used as input to the Wilcoxon Signed-Rank test. The optimally performing classifier for EER is defined as the classifier with the lowest average EER (i.e., 5-NN (Man) with 19.50%), and the optimal classifier for AUC is defined as the classifier with the highest average AUC (i.e., 5-NN (Man) at 88.49%). The two best classifiers were compared to each other classifier, thereby creating four separate hypothesis tests for each error rate. The null hypothesis is that there is no difference between the performance of the best performing classifier and all other classifiers. If the null hypothesis cannot be rejected, then the performance of the classifier in question is at least as good as the top-performing classifier and therefore must also be considered top-performing. The result of calculating the statistical significance of the best classifier for both AUC and EER compared to all other classifiers was that the best classifier was not statistically significantly better than any of the other four classifiers being considered (see Table 4.4). This is not an unexpected outcome given the small amount of data and the fact that statistical significance is strongly influenced by the size of the dataset. These results for statistical significance also support the need for a larger study of this type.

Metric	**Classifier**				
	Naïve Bayes		Decision	5–NN	
	Gaussian	Kernel Density	Tree	Euclidean	Manhattan
EER (%)	24.02 (1.0000)	20.25 (0.9375)	31.77 (0.7810)	26.08 (0.2969)	**19.50**
AUC (%)	83.35 (1.0000)	85.36 (1.0000)	73.80 (0.0781)	79.94 (0.2969)	**88.49**

Table 4.4: EER and AUC medians for all classifiers. The bolded numbers represent the best classifier based on either EER or AUC, and the number in brackets after each percentage is the result of the Wilcoxon Signed-Rank Test, ρ. The bolded EER and AUC values were not considered significantly different than the other classifiers at the $\alpha = 0.05$ level.

4.5 Study Limitations

There were several potential limitations in this study. The results of this experiment should be assessed with these biases in mind.

Devices: The devices used in the study were provided by the study participants. This means that the experimenter could control neither the services and apps on each device, nor

whether the device owner chose to upgrade their device operating system during the study. The participants were asked not to upgrade during the study, but no information is available to ensure this because it was not tracked beyond the initial question regarding current operating system asked at the start of the study.

Display Differences: The iPod Touch display has fewer pixels than that of the iPhone 4, which may affect typing patterns. The iPod Touch 3rd generation and iPhone 3GS display is 480 x 320 pixels; the iPhone4 display has 960 x 640 pixels.

Realistic Conditions: Some realism was sacrificed in typing practices in order to protect the participant's privacy. Specifically, the participant was asked to use a custom-designed application rather than the real email and text messaging apps so that they could be certain that only keystrokes in the custom application were being sampled. The lack of realism could mean that the participant typed non–typical words and phrases, or typed in a way that they would not in normal practice in order to participate fully in the study.

Features Limited: Auto–capitalization and auto–correct (for spelling) were disabled in the custom application. This also limits the realism in the typing environment because the participant must either correct mistakes or leave typing mistakes as they are. It also means that the backspace key frequency, which appeared in nearly all of the participants' most-frequent lists, is artificially high. However, the backspace frequency is not an issue in further studies because it may simply be replaced by a more frequently used keystroke. Further studies should, however, allow the use of the auto–capitalization and auto-correct features to determine whether this has an effect on the distinctive nature of typing patterns.

Language: The participants were asked to type in English only, which may not represent their usual typing patterns. It is possible that, despite all participants being native English speakers, that they prefer to type in another language or an English variant such as text-speak. Since both were disallowed during the study, this may affect the device owner's usual typing pattern. The reason for this requirement was that initial examination of the data gathered in this study used the standard frequency of English language characters to select which timings to use in the feature vector. However, it was found during the study that these were not the most effective and discriminatory values. Future studies need not limit the language used in typing, and should examine the effects on results of allowing the use of auto-capitalization and auto-correct features.

Participant Skill Level: While every effort was made to include participants from a variety of skill levels, it is possible that an individual participant's skill level could have

an effect on their typing patterns and thus on the study results. The study lasted approximately three weeks, so it may be the case that the typing patterns of some of the less skilled participants could have changed rather dramatically during the study timeframe. Another consideration related to participant skill level is the perceived difference in typing patterns between a skilled and unskilled soft keyboard typist. In general, an unskilled typist may be slower and potentially tap the keys for a longer period of time compared to a more skilled typist. The former difference would increase the inter-key latency times and the latter would increase the key hold times for unskilled typists, making their patterns more clearly distinguishable from skilled typists. These differences could affect error rates by making them abnormally low for the under-represented group. For instance, if there were far more unskilled than skilled typists, it would be the skilled typists' error rates that would be abnormally low, and vice versa. This bias can be minimized by selecting a wide range of skill levels and ensuring that the number of participants at a given skill level is approximately equal.

Small, Unbalanced Datasets: The number of participants in the study was small ($N = 8$) and thus the amount of data gathered as part of this experiment was also small. This is an acceptable limitation since this is intended to be a feasibility study upon which a larger study may be based in the future.

4.6 Keystroke Dynamics in the Transparent Authentication Framework

Keystroke dynamics is intended to be one of two biometrics used in the Transparent Authentication Framework presented in this research, although the Framework is intended to support as many biometrics as the designer may wish. The results of this study have shown that keystroke dynamics hold promise as a potential biometric for use with authentication services. The error rates are acceptable, but they are not low enough to use as the sole biometric or authentication tool, particularly in light of the ranges of error rates cited by other similar studies. Considering that the keystroke dynamics are intended to be combined with speaker recognition results into a multimodal biometric, the results shown here are sufficient to support the use of keystroke dynamics in this manner. Also, the intended transparent nature of the authentication framework implies that the biometrics used should be sampled while the device user goes about other common tasks on the device. Keystroke dynamics is ideal for this purpose, and the promising results shown in this study support its continued use as a biometric used to support authentication on mobile devices.

4.7 Summary

This chapter has introduced the Keystroke Dynamics Feasibility study, the first purpose of which was to collect keystroke dynamics information from users of Apple devices with soft keyboards and secondly to determine which of five possible pattern classification algorithms is best suited to classifying the timing data gathered from the study participants. The study's design was detailed, and the results and analysis thereof were presented. The outcome of this study is that its error rates, which are similar to error rates seen in other studies that used different keyboard types, suggest that keystroke dynamics is not sufficiently distinctive to use as the sole basis for determining whether the user is the device owner, although it is sufficiently distinctive to consider combining it with one or more other biometrics in order to improve accuracy. The results support using the inter-key latency rather than key hold time, since the former has been shown to be more distinctive both in this study and other similar studies. However, the results of the study do not support the choice of a particular classifier due to the high variability in error rates, and the opposing recommendations when examining the FAR and FRR data versus the EER and AUC data. Finally, the study's results, although promising, were based on a small group of participants ($N = 8$); therefore, they should be verified via a larger study of the same or similar design that also uses a soft keyboard.

Chapter 5

Speaker Verification Feasibility Study

This chapter presents the study[1] that was carried out to determine whether speaker verification is a sufficiently discriminatory biometric to contribute to identity verification in the Transparent Authentication Framework. The chapter begins with an outline of the study goals and justification for the voice pattern corpus creation, then continues with a description of the study design including participants and materials used. The study methodology is then described in detail. Finally, the results of the study are presented and the study limitations discussed, and a reflection of the relevance of the study within the context of the Transparent Authentication Framework is provided.

5.1 Study Goals

This study was designed to answer three research questions:

1. Is there sufficient distinctive information in voice patterns to verify the identity of the device owner, and to justify using it as a biometric in the Transparent Authentication Framework?

2. Are the error rates produced by the five pattern classifiers low enough to support identity verification on a mobile device?

3. Can an optimal classifier be chosen for the speaker verification behavioral biometric, based on the data gathered during this study? "Optimal" is defined as the classifier with the lowest error rate that also respects the mobile device environment's limitations in processor speed and available memory.

[1] University of Glasgow ethics approval number CSE00977

Answers to these research questions provide a partial response to Hypothesis H1 as stated in Chapter 1.4.1, along with the answers provided by the Keystroke Dynamics Study described in Chapter 4.

Answering these questions required a corpus of voice patterns from mobile devices, specifically from the Apple iPhone and iPod Touch. This corpus had the following requirements:

Type: The data type of the recordings must be known, and must be of a type that can be processed by the feature extraction software described in Section 5.3. There is a wide range of choices that may be used with this software, so this requirement is not overly restrictive.

Recording Quality: The recording quality must not only be known, but be high enough to support voice pattern classification. Audio pattern classification research suggests that the minimum recording quality for voice-only recordings is 8000 Hz since most audible sounds in speech occur in the frequency range between 250 Hz and 8000 Hz [118].

Gender Mixture: In order to avoid possible bias in the case that the corpus contains only one gender of speaker, a mixture of genders should be represented.

Single Speaker: The purpose of the voice classification portion of this research is to verify that the person speaking into the device is the device owner. Therefore, since the most usual scenario for this is during a phone call, only a single voice will be gathered by the device microphone (or attached microphone in the case of the iPod Touch) and thus it is reasonable to use only single speaker recordings in the corpus.

Device: At a minimum, the corpus should represent telephone conversations rather than other voice recordings. This study focuses on the feasibility of speaker verification on the Apple iPhone and iPod Touch, so the speaker voice samples should ideally be gathered from these two devices only in order to avoid bias.

Variety: The speech gathered must not be limited to a specific word or phrase since this study's model is text–independent. Furthermore, in order to accurately represent the possible circumstances in which voice data may be gathered (e.g., amount and type of background noise, while the speaker is moving) the voice samples should be collected in an uncontrived way.

Amount of Data: Text–independent speaker verification systems require a large amount (e.g., several minutes) of training data in order to achieve acceptably low pattern classification error rates [70]. Therefore, the corpus used must contain a large amount of sample data for each participant.

Several corpora meet some of these requirements, including the SWITCHBOARD corpus [121] and the Fisher corpus [157]. The SWITCHBOARD corpus is a good choice for voice data studies because it contains a large amount of data from telephone conversations, contains samples from both male and female speakers, and the quality and type of each recording is known. However, these samples are not from an Apple iPhone or iPod Touch, or any smartphone for that matter. Furthermore, they do not have the variety in location during the call that is required for this study, since the speakers are all using landline telephones and are therefore at a fixed location that may or may not have background noise. The use of landline telephones also may mean that the subjects were not moving while speaking, which may be different from mobile device users. Furthermore, the samples in this corpus represent conversations with two or more people, which means that the samples would have to be pre–processed to extract a single speaker. These limitations make the SWITCHBOARD corpus a poor corpus to support this voice study.

The Fisher corpus also contains conversations rather than single person recordings, and was built for *speech* recognition rather than *speaker* recognition, meaning the *words spoken* were important rather than *who* spoke them. As with the SWITCHBOARD corpus, the Fisher corpus is composed of telephone conversations but not from the iPhone or iPod Touch, making it a poor choice for this study. There are several other similar subscription–based corpora available such as CALLHOME [158] and CALLFRIEND [159] but each were poor choices for this study for similar reasons to the other two corpora considered above, in addition to the fact that each of these corpora cost USD $1500. The MIT speaker verification corpus [160] is a possible candidate for use with this study, although it is intended for text–dependent speaker verification. Since a corpus that meets all of the needs of this research was not already available, and since this feasibility study did not require a large number of individual speakers, an application was built for the Apple iPhone and iPod Touch that would serve to gather the required data. This application is described in the following sections.

5.2 Study Design

This study has two parts: the first part was data gathering, in which voice recordings were gathered from participants using a custom iPhone and iPod Touch application, and the second part involved presenting the data gathered to software used to create the feature vectors common in voice classification studies, and next to a series of pattern classifiers. This latter step is used to determine whether there is enough distinguishing information in the voice patterns gathered to verify the device owner's identity, and thus provide an answer to the three research questions described in Section 5.1.

A between–groups design [152, p. 74] was used for the data gathering part of this study. Each

participant was in turn considered the *owner* of their device and the remaining participants made up the group *rest–of–world*. Each of the participants used either an iPhone or an iPod Touch during the experiment. All devices had a single user, and recordings made were of that user and no one else. The participants were instructed to not record conversations – their voice should be the only voice on the recordings. The data gathered was then presented to the pattern classifiers chosen as part of the second part of the study.

The dependent variables were the classifier error rates, which depended on the recordings and the feature vectors created from them. This study's independent variable was which classifier was used on which dataset, which depended in turn on which participant was considered the device owner.

5.2.1 Participants

There were nine participants involved in the data gathering part of this study. They ranged in age from the 21–29 age range to the 50–59 age range, and had owned their device between less than one year and up to three years. They ranged in experience level from novice (two participants, had never used an Apple device before this experiment) to intermediate (three participants, up to two years experience using an Apple device) and expert (four participants, more than two years experience with an Apple device). The participants represented a wide range of accents including English, Scottish, Canadian, American and South African. There were five female and four male participants, each of whom received a Consent Form and Information Sheet. The participants agreed to participate in the study via a web page and gave their consent to participate by choosing to tick a checkbox on the sign-up page. The participants were volunteers who supplied their own mobile device and were not paid in any way for their participation.

5.2.2 Apparatus and Materials

The participant–owned devices used were one iPhone 4, three iPhone 3GSes, three iPod Touches 3rd Generation, and two iPod Touches 2nd Generation, for a total of nine devices. Microphones were not provided to the participants; they used either the built–in microphone or a peripheral device. The iPod Touch owner participants provided their own microphone since these devices do not have one built in; many used one that was attached to headphones. They type of microphone used was not captured for this study, nor was whether the same microphone was used for all recordings. All the devices had iOS 4.2 or better as the operating system and most had iOS 5.0.1. The participants were instructed to use the devices in any location, to follow their normal use patterns, and to speak in English. They were also reminded not to record others' voices, but that regular background noise was acceptable.

Part 2 of the study, pattern classification, used SPro version 4.0.1 [2] for feature vector creation and MatLab version R2012b with the standard pattern classification algorithms included in MatLab's Statistics Toolbox for pattern classification, error rate calculations, and statistical significance testing.

SPro output is intended for use by a software program called ALIZE [161], which is an open–source biometrics authentication platform created by researchers at the Laboratoire Informatique D'Avignon at the University of Avignon. ALIZE was tested for use with the recordings gathered during the speaker verification study, but MatLab was considered more convenient for use in this research, despite ALIZE's intended use with SPro.

5.2.3 Procedure

Part 1: Data Gathering

Part 1 of the study used a purpose-built iPhone application to record voice patterns from the study participants. The Apple iPhone and iPod Touch both have a voice memo application, but it was decided that it would not be used for this study in order to protect the user's privacy. By using a completely separate application, the user must choose to provide their voice recordings for this study; by using an existing application such as Voice Memo, the user may be suspicious that other voice–related features, such as making a telephone call, may be recorded as well. Furthermore, creating a separate application allowed for finer–grain control of the data acquisition since the data type and format could be controlled. The trade-off is that the VoiceData application does not run in the background as would be required for a transparent authentication method, although this was not considered a limitation of this study since its purpose was simply to gather voice recordings.

The participant pool was expanded by automating many of the processes related to study sign-up. The information sheet was sent to each participant electronically, and then each participant filled in a web form hosted on the experimenter's university website. The web form contained questions about the participant (name, age, email address and amount of experience with their device), and questions about the device they intended to use for the experiment (type, length of ownership, UDID). The final question asked whether the participant had read the information sheet and whether they agreed to participate in the study – the text was the same as that above the signature line in the consent form. If the participant checked "Yes" next to this question, it was considered equivalent to signing the consent form. The web form was created using an online form creator called JotForm[3] and used Dropbox[4]

[2] http://www.irisa.fr/metiss/guig/spro/spro-4.0.1/spro_1.html
[3] http://www.jotform.com
[4] http://www.dropbox.com

to collect the participant data. All data, once submitted, was removed from both websites to preserve participant privacy.

Once the initial study sign–up was complete, the participant was sent an email containing the VoiceData application and instructions on its installation and use. After installation, the participants used the custom application to record their voices as often as they wished. After three weeks, the participants sent recordings via email to the experimenter by using the built–in email function in the application. After data gathering was complete, another email was sent to participants with instructions on how to remove the VoiceData application from their device.

Part 2: Pattern Classification

The gathered data was pre–processed and converted into feature vectors. It was then presented to five pattern classifiers available in MatLab version R2012b. The pre-processing and feature vector creation methods are described in Section 5.3.3.

In order to answer the research questions stated in Section 5.1, five pattern classifiers were tested during this study: k-Nearest Neighbor with both Manhattan and Euclidean distance measures (k-NN (Man) and k-NN (Eucl), respectively), Decision Tree (DT), and Naïve Bayes with Kernel Density (NB(KD)) and Gaussian (NB(Gau)) estimations. The data gathered in the first part of the study was divided into owner and rest-of-world patterns, as was done in the Keystroke Dynamics study. In this case, there were nine participants, each of whom was considered the device owner in turn, while the remaining participants became part of the rest-of-world group. This resulted in nine datasets, each with an owner and rest-of-world group. Unlike the Keystroke Dynamics study, however, the only difference between each of the nine datasets was the classification of each feature vector into either the owner or rest-of-world class; otherwise, the data remained the same. In other words, the data in each of the nine datasets is the same; it is only the classification of particular feature vectors as belonging to the owner that changed between the nine datasets.

Each classifier was trained using supervised learning methods, similar to the Keystroke Dynamics study. However, the large amount of data gathered meant that each data run through the classification algorithms took a significant amount of time (about seven hours per run), so the dataset was partitioned into randomly chosen sets of 5000 patterns evenly divided between owner and rest-of-world patterns. 10-fold cross-validation methods were used to verify the classifier accuracy; each cross-validation exercise used a different set of 5000 patterns. The classification problem for this study was two-class; either the voice pattern belongs to the owner of the mobile device, or it belongs to the rest-of-world class.

5.3 Data Acquisition

The first prototype for the data acquisition application used Apple's CoreAudio libraries, which are written in the C programming language. Testing revealed issues concerning the use of the application. For example, standard functionality such as fast forward and rewind did not work as expected. The second prototype used the AVAudioSession libraries, which are a set of wrapper classes for CoreAudio that simplify adding standard functionality such as fast forward, rewind and playback resumption.

(a) Initial screen with privacy warning.

(b) Main screen.

(c) Playback screen.

Figure 5.1: Screenshots of the VoiceData application used to gather the data required for the speaker verification feasibility study.

The application interface is shown in Figure 5.1. Figure 5.1a shows the initial screen upon application launch. An alert box was presented each time the application entered the foreground; it warned the user about the privacy limitations inherent in a study of this type. Its intention was to ensure that the user knows that they cannot expect their recordings gathered with the application to remain completely private, and was included in order to aid with obtaining ethics approval for the study. After tapping the *I Understand* button on the alert box, the main application screen is revealed, as seen in Figure 5.1b. The interface contains Record and Stop buttons for beginning or ending recording and a counter that shows the amount of time that has passed since recording began. The counter also served as a visual reminder to the user that recording is taking place. The other two buttons on this screen are to send the

data to the experimenter via email, and to play recordings made using this application. Tapping the *Play...* button brings up the screen pictured in Figure 5.1c. All previously recorded data is shown in the list, and tapping on one begins playback of that audio file. The slider at the bottom of the screen shows the progression of the playback and may also be used to fast forward or rewind the recording by sliding the indicator to the left or right (this is known as *scrubbing*). The Back button in the top left corner returns the user to the previous screen (Figure 5.1b).

The audio playback feature was not required to fulfill the purpose of the application, which was to provide an interface for gathering audio data recorded on an iPhone or iPod Touch. However, it was included as a way of providing a more well–rounded interface that could be used as a voice memo application, and was included to encourage the participant to record more data. Whether or not this inclusion yielded more data was not examined as part of this study.

5.3.1 Data and File Formats

The recorded audio files were stored on the participant's device, so keeping the file size small was an important goal when designing the VoiceData application. Two formats govern audio data storage: *file format* and *data format*. The file format is the container that stores the audio information itself; the data format specifies how this stored audio information is encoded. One file format may be able to store many data formats: for instance, a .caf file format may be used to store audio data encoded with Linear PCM, Advanced Audio Coding (AAC) or MP3 data formats, amongst others. The file format has little effect on the stored data file size, thus .caf was chosen because it is native to Apple iOS development.

The data encoding format, however, has a large impact on file size. Initially, the application was designed using the LinearPCM data format enclosed in a CoreAudio File (.caf) file format. This resulted in a 16000 Hz, 15 second long audio file that was over 1 MB in size. Since participants were encouraged to record significantly longer recordings than 15 seconds, the audio data may have taken a large portion of their available device memory. The AAC data format, a successor to the MP3 file format (and thus a lossy representation), produces a file that takes less disk space but at the cost of lower quality audio. The same 16000 Hz, 15 second audio file occupied just 61 KB. While many audio recordings that focus on voice data are recorded at 8000 Hz [118, 132], 16000 Hz was chosen for this study because it may help limit the lower quality audio effects that using the AAC data format entails (this assumption was not tested).

5.3.2 Data Retrieval

The concern regarding using too much space on the participant's device was further alleviated by sending the data to the experimenter in chunks of 5MB or greater and removing the sent data from the participant's device. To enable physically distant participants to submit data, a facility for emailing the data to the experimenter was created. The data consisted of many potentially large files, thus attaching them to a single email was infeasible since they may be too large to be sent or received without error. To combine the audio files into a single file, a compression program was used to create a single zip file that contained all of the recorded audio files and attach this file to the email.

Several other methods were considered for data gathering, including uploading to a webserver or to a facility such as Dropbox. These methods required either significant participant effort and technical knowledge (Dropbox uploads) or complicated and lengthy setup by the experimenter (server uploads). Since the devices tested have email capabilities, and most participants had email set up and knew how to use it on the device, the email option provided a convenient choice that worked very well in practice. Just over 57 MB of audio data was gathered during this study.

5.3.3 Feature Extraction

Some of the considerations that may affect voice audio feature extraction are the amount of noise and the overall quality of the recording, as well as whether the system is text–dependent or text–independent. Speaker verification research focuses heavily on the choice of specific audio data features that may be more or less deterministic, in terms of verifying the speaker's identity [119]. These features include physical traits such as vocal tract shape and larynx size, which are often combined with behavioral traits such as accent and pronunciation to improve recognition accuracy [119]. Since this work is not intended to push the boundaries of speaker verification research, established methods were sought to create a feature vector that would suit both the data and the *verification* of the speaker's identity. To render the data suitable for these purposes, it was necessary to perform pre–processing and feature extraction on the data gathered for this study. Common pre–processing tasks for audio data include removing background noise and periods of silence, and filtering out high–frequency sounds that are outside the range of human hearing.

The tool used for preprocessing was SPro [162, 163], which is a freeware signal processing toolkit that focuses on audio recordings that contain speech. It provides feature extraction algorithms that are common to speech-related audio applications, specifically those required for speaker recognition and verification. It is written in the ANSI-C programming language and runs via a command-line interface. SPro is a fully-featured speaker verification tool, and

includes such speech analysis capabilities as variable resolution spectral analysis, filter-bank analysis, linear predictive analysis, and cepstral analysis. The SPro manual [162] is a good source of information on these techniques for the interested reader, as is the reference text by Homayoon Beigi [164] and the papers by Kinnunen *et al.* [119] and Bimbot *et al.* [69].

Data gathered in the speaker verification study was processed using cepstral analysis because it is a popular and effective method for speech recognition, and thus is also often used in speaker verification work. Bimbot *et al.* call cepstral analysis "...the most commonly [sic] speech parameterization used in speaker verification..." [69, p.1], and this assessment was used to justify the use of cepstral analysis as the appropriate analytical method for this research. Essentially, cepstral analysis is used to create a feature vector that has the most distinctive parts of speech emphasized in order to increase its usefulness as a biometric.

SPro takes in as input a *waveform stream* (the input recording) and outputs a *feature stream* that contains the feature vectors of the input recording. Feature streams are output files in an SPro-specific file format (*.mfcc* extension) that includes header information about the input audio recording. The SPro command for performing the cepstral analysis is *sfbcep*, as shown below.

```
sfbcep --format = 'PCM16' --sample-rate = 16000 <infile>.caf <outfile>.mfcc
```

Using MatLab, and specifically the classifiers used in other studies undertaken as part of this research, would provide results that can be more easily compared to the results of the Keystroke Dynamics study, as seen in Chapter 4. The additional control over the output of the classifiers also provided a more flexible environment for combining classifier output as is described in the Multimodal Biometrics study chapter (see Chapter 6).

The MatLab input data was formatted as a standard ASCII text file that was separated into rows and columns using whitespace. The standard SPro output can be converted from .mfcc files to simple ASCII files via the command *scopy*, which resulted in rows and columns of feature vector values, where each row represented the features extracted from a 20ms long voice sample, and the columns represented the individual features selected by SPro. The *scopy* command is shown below:

```
scopy -o ascii <input filename>.mfcc <output filename>.txt
```

After feature extraction, the just over 57 MB of samples gathered were converted to more than 1.8 million individual feature vectors with 12 features each. The data was manipulated into different sets for each study participant, each of whom in turn was considered the device *owner* and the other participants were considered part of the *rest–of–world* class. This rest-of-world set is often called a *world model* or *cohort model* in speaker recognition and

verification research [119]. Irrespective of its nomenclature, the feature vectors included in this set were used as negative (i.e., non–owner) samples during training, and were thus used to create a universal background model during pattern classification.

The large number of feature vectors created by SPro became problematic when presented to the chosen classifiers. A large amount of data must be stored in a matrix during training and testing, which uses a large amount of memory. Furthermore, the k–NN classifiers must compare each testing data point to each training data point individually, a task that increases the time taken to classify exponentially with the addition of each new training point. As a result, the classifiers took far too long to train and classify the data, so the decision was made to reduce the amount of data to a more manageable level. After testing several training and testing set sizes for both speed and differences in the classifier accuracy, the training set size contained 4000 patterns and the testing set contained 1000 patterns. The entire dataset was not used in this case, although each of the 10 cross-validation runs used a different set of 5000 randomly chosen patterns in order to use as much of the data as possible. The training and testing sets were balanced; each contained the same number of owner and world patterns.

5.4 Results and Analysis

Table 5.1 shows the error rates produced when the speaker verification feature vectors were presented to each of the five classifiers. The error rates include False Accept Rate (FAR), False Reject Rate (FRR), Equal Error Rate (EER) and Area Under Curve (AUC), where the curve in question is the Receiver Operating Characteristic (ROC) curve. For the first three error rates, lower rates indicate fewer errors. For AUC, a higher rate indicates fewer errors.

There was a large discrepancy between the amount gathered from each participant, as can be seen in Figure 5.2. This discrepancy has significant implications for analyzing the results of the pattern classification on this data. In general, the lack of balance between positive data samples (i.e., those of the device owner) and negative data samples (i.e., those of the rest-of-world population) means that FAR and FRR are less useful in determining the quality of the pattern classifier being tested. This has to do with the fact that FAR and FRR (and crude accuracy) are sensitive to differences in the base rate of the positive and negative classes. This means that if there are far more of one type of class than the other, then a few misclassifications in the under-represented class will make a large difference in the calculated error rate, be it FAR, FRR or crude accuracy [134]. This can be used to explain overestimation in the accuracy of a particular classifier, although it is difficult to detect such an overestimation unless study results are verified with a similar study. Since EER is related to FAR and FRR, then it too falls to possible estimation errors with unbalanced datasets. The area under the receiver operating characteristic curve (AUC-ROC) represents the probability

of a true positive response from a given classifier in a binary problem, and thus is not affected by unbalanced datasets [134]. Due to these constraints, the data was balanced into sets containing 5000 patterns each: 2500 from the owner and 2500 from the rest-of-world group.

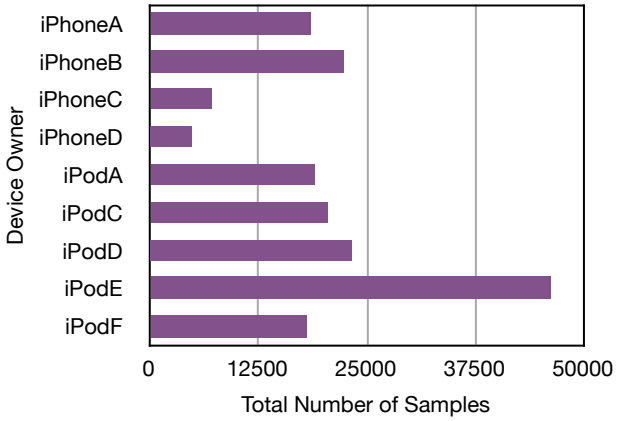


Figure 5.2: Amount of voice data gathered, by owner.

The results of pattern classification, shown in Table 5.1, are the median values of the 10 cross-validation runs. The initial results from the five pattern classifiers show medium low FAR and FRR values, slightly better EER values, and very good AUC values. Speaker verification studies for telephone conversations generally have EER values of approximately 10% [69, 71], although this is dependent on factors such as amount of background noise and amount of training data. The EER values in Table 5.1 are somewhat higher than this, which is likely due to the limits on training data. Since the datasets used for each cross-validation step were balanced in terms of owner and rest-of-world patterns, the FAR and FRR values are within the same range, although they are higher than expected. These higher levels can be attributed to the natural sources of variance in audio data used for pattern classification, and the fact that this study used a text-independent model. Such a model does not depend on the participant speaking a particular word or phrase that has been chosen at enrolment. Instead, the participant speaks freely and comparisons are made between individual utterances that are very likely to be different both in content and in location, and thus inter-utterance differences in channel clarity and quality may exist. Text-independent models are known to have higher error rates than text-dependent systems [118], which may explain the higher error rates in the results of this study.

The FAR and FRR values show an acceptable level to justify a larger study (i.e., with more participants rather than with more data). The variation between classifier results on a per-owner basis was low; for example, the FAR values for iPodA ranged from a low of 33.80% to a high of 39.70%, which means that all classifiers tested showed similar capability to classify this owner's speaker patterns. In other cases, such as for iPhoneA, the FAR and FRR values

Result	Owner	Classifier				
		Naïve Bayes		Decision	5–NN	
		Gaussian	Kernel Density	Tree	Euclidean	Manhattan
FAR (%)	iPhoneA	52.10	42.80	42.20	40.80	41.90
	iPhoneB	43.50	37.30	37.30	32.90	33.80
	iPhoneC	44.90	40.40	35.70	27.80	28.80
	iPhoneD	28.60	29.90	35.10	32.10	33.10
	iPodA	39.70	36.00	34.90	33.80	35.30
	iPodC	46.10	45.30	37.30	37.80	39.70
	iPodD	20.90	21.20	33.80	24.80	23.60
	iPodE	28.80	28.50	35.30	27.80	28.30
	iPodF	44.60	35.40	31.10	29.10	30.50
FRR (%)	iPhoneA	26.90	38.20	42.40	32.20	32.90
	iPhoneB	29.50	35.20	36.30	24.40	25.00
	iPhoneC	24.40	27.50	35.80	27.60	27.40
	iPhoneD	49.30	41.50	35.40	24.10	23.80
	iPodA	34.10	32.80	34.00	19.40	20.00
	iPodC	25.80	26.30	37.20	21.90	22.10
	iPodD	43.10	42.70	34.50	30.60	31.20
	iPodE	34.80	35.50	35.00	26.00	26.00
	iPodF	17.80	25.10	31.40	19.60	19.10
EER (%)	iPhoneA	39.50	40.20	42.12	37.12	38.06
	iPhoneB	36.80	36.30	36.87	29.33	29.55
	iPhoneC	34.50	32.80	35.50	27.93	28.54
	iPhoneD	37.30	35.30	36.14	28.61	28.94
	iPodA	35.10	34.50	34.47	28.68	28.48
	iPodC	35.60	33.70	37.82	29.99	31.39
	iPodD	31.90	31.80	34.30	27.43	27.72
	iPodE	31.50	32.20	34.60	27.25	27.65
	iPodF	28.30	29.70	31.44	25.07	25.85
AUC (%)	iPhoneA	64.36	63.61	59.63	68.15	67.14
	iPhoneB	68.08	70.65	65.83	77.75	77.24
	iPhoneC	71.55	73.24	67.32	78.42	77.94
	iPhoneD	66.36	69.37	66.00	77.90	78.36
	iPodA	70.68	71.70	68.46	79.33	79.35
	iPodC	70.67	72.00	65.52	76.63	75.71
	iPodD	74.93	75.55	68.55	79.63	79.48
	iPodE	75.85	75.59	68.52	79.66	79.06
	iPodF	77.78	77.45	71.64	82.50	81.62

Table 5.1: Pattern classifier results, as median values of 10–fold cross-validation. Rest-of-world participants are repeated in both training and testing sets (data is not repeated).

are among the highest for all classifiers when compared to other owners. This suggests that iPhoneA's owner was harder to distinguish from other speakers and that there also may be people for whom speaker verification is not a suitable biometric. Such a failure-to-enrol is common with all types of biometrics. Despite their unacceptable levels, the FAR and FRR values suggest that the optimal classifier for this data is either of the two 5-NN classifiers, although neither show low enough error levels to justify using speaker verification as a *sole* biometric. It should, instead, be combined with another biometric to improve the error rates and increase the likelihood of correctly identifying the device owner.

AUC values ranged from an overall low of 59.63% for iPhoneA using the DT classifier to a high of 82.50% for iPodF using the 5-NN (Eucl) classifier. Such a large range indicates that most of the classifiers performed reasonably well for most owners, but that more training data was required. The results on a per-owner basis were more regular: iPhoneA, which had the lowest (and thus the worst-performing) AUC values ranged from a high of 68.15% to a low of 59.63%, which means that iPhoneA's patterns, as seen with the EER values, are not very distinctive. The classifiers were only somewhat better than chance, which has an AUC value of 50%. On the other hand, the AUC values for iPodF are higher while still maintaining a similar variation to those of iPhoneA. AUC values for iPodF ranged from a low of 71.64% for DT to a high of 82.50% for 5-NN (Eucl). These values are within acceptable ranges for use as a biometric classifier, although not in production systems. The AUC values overall suggest that the optimal classifier for this data is either of the two 5-NN classifiers, as was the conclusion when examining the other error rates.

The classifiers used were not tuned or tweaked in any way when performing these classifier comparisons. The typical workflow when considering pattern classification algorithms for a particular task is first to train a model, then measure the classifier's accuracy with test data. The results of accuracy measurement are used to determine what, if any, steps may be taken to simplify the model and thus reduce the amount of data needed to maintain that level of accuracy. In some cases, model simplification may also lead to a more accurate model. This study was a feasibility study to determine whether potentially noisy data gathered from mobile device owners contains enough information to verify the device owner's identity. As such, the results of this study support a larger–scale experiment that can include classifier model tuning. Such steps may lead to an increase in the classifier accuracy and a reduction in the associated error rates.

Statistical Significance

As with the keystroke data analysis, there are two requirements that the statistical significance test must meet, as follows:

1. Non-Parametric: the data gathered is not parametric (i.e., does not follow a normal or Gaussian distribution) as evidenced both by the fact that the EER and AUC values range between 0 and 1 and the results of the Kolmogorov–Smirnov test [152] for each of the calculations in turn, as shown in Figure 5.2.

2. Pairwise tests: For each owner data, the comparison is between the AUC or EER for each classifier – this means that the results for the optimal classifier should be compared to the AUC and EER for each of the remaining four classifiers in turn, which implies using a test that will compare pairs of values.

The AUC and EER data did not follow a normal distribution, as shown by the results of the Kolmogorov–Smirnov tests seen in Table 5.2. The values in each case are well below the $\rho < 0.05$ levels required for Gaussian distributions. This implies that the data is non–parametric in nature. Thus, a non–parametric test such as the Wilcoxon Signed–Rank Test is suitable for this data, as it was for the keystroke data in the previous chapter.

Metric	**Classifier**				
	Naïve Bayes		Decision Tree	5–NN	
	Gaussian	Kernel Density		Euclidean	Manhattan
EER (%)	0.0009	0.0008	0.0007	0.0013	0.0012
AUC (%)	0.0000	0.0000	0.0000	0.0000	0.0000

Table 5.2: Results of the Kolmogorov-Smirnov test of distribution shape for EER and AUC values. Since each value is less than the $\alpha = 0.05$ significance level, then each distribution of data is considered significantly different from a normal distribution (i.e., the distribution is non–parametric).

Metric	**Classifier**				
	Naïve Bayes		Decision Tree	5–NN	
	Gaussian	Kernel Density		Euclidean	Manhattan
EER (%)	35.10 (0.0039)	33.70 (0.0039)	35.50 (0.0039)	28.61 (0.0078)	**28.54** –
AUC (%)	70.68 (0.0039)	72.00 (0.0039)	67.32 (0.0039)	**78.42** –	78.36 (0.0273)

Table 5.3: EER and AUC medians for all classifiers. The bolded numbers represent the best classifier based on either EER or AUC, and the number in brackets after each percentage is the result of the Wilcoxon Signed–Rank Test, ρ. 5-NN (Eucl) was considered significantly better than the other classifiers for EER and AUC at the $\alpha = 0.05$ level.

Table 5.3 shows the results of the Wilcoxon Signed–Rank test on the AUC and EER values for the speaker verification data. The classifier with the lowest EER median value, 5–NN

(Man) with 28.54% median EER, was chosen as the optimal classifier; this is shown by the bolded median value in the table. 5-NN (Man) was then compared to each other classifier using the Wilcoxon Signed–Rank test, with the results showing that 5-NN (Man) was statistically significantly better than each of the other classifiers at the $\rho < 0.05$ significance level. The table shows the same values for the first three classifiers because when comparing each value for each owner to that of 5-NN (Man), the latter is the lower value in every case. The Wilcoxon Signed-Rank test assigns either a plus or minus sign to each comparison depending on whether the optimal classifier has a higher or lower value than the number to which it is compared. Therefore, each comparison was assigned the same sign, and the probability of seeing the median value reported does not change from one column to the other. The optimal classifier according to the AUC data is that with the highest AUC value; in this case it was the 5-NN (Eucl) classifier with median AUC over all owners of 78.42%. The results of the Wilcoxon Signed–Rank tests that compared 5-NN (Eucl) to all other classifiers showed that 5-NN (Eucl) was statistically significantly better than the others. The results of the statistical significance justify the choice of either of the two k-NN classifiers as the optimal one for the speaker verification data. These results should be verified with a larger study with more participants before implementing speaker verification as a biometric in the Transparent Authentication Framework.

5.5 Study Limitations

There were several possible study limitations that should be kept in mind when assessing the results of this experiment. They are as follows:

Study Size: As with the Keystroke Dynamics Study, the size of the speaker verification study was small ($N = 9$), which means that any results from the study may not be as conclusive as the results from a larger study. The amount of data gathered was significant, but only from a few participants which means that inter-participant differences may not have been significant enough to allow the owner to be distinguished from the rest-of-world patterns. This limitation can be alleviated by repeating this study with a larger number of participants.

Devices: Although all devices used were Apple iPhones and iPod Touches, the devices belonged to the study participants so the amount or type of applications on the device could not be controlled. Similarly, updates to the operating system may have been made during the experiment, which may have subtly changed the study application's functionality. The amount of available space and processor speeds of each device is not known, although this is unlikely to have had an impact on the study since all data reached the experimenter as expected.

Background Noise: Sounds other than the speaker's voice may have been recorded, and since the data was not filtered explicitly (although it may have been done as part of the feature extraction step by the SPro software), these extra sounds may cause differences in patterns from the same speaker. This is considered an acceptable limitation because it more realistically mimics the usual environment during a phone call – it is unlikely that the device owner will always be in a completely silent area during phone calls.

Speech Differences: Several speakers mentioned casually to the experimenter that they read aloud books or other text during recording in order to gather more data. This may affect their natural voice pattern since they are not speaking their own words with their own cadence and inflection, as it is expected they would if they were on a phone call or recording a voice memo. In terms of this study, it is an acceptable limitation because their voice pattern would be from this same source throughout their recorded voice samples.

5.6 Speaker Verification Accuracy

Speaker verification accuracy is affected by three major sources [119, 165]:

1. Phonetic variability: Differences between the words and phrases as spoken in the training set and in the testing set. This is a significant source of loss of accuracy in this study since a text-independent model was used, which implies that the words and phrases are very likely different between the training and testing sets.

2. Technical factors: these include the nature of the channel used, microphone quality, recording quality, and data loss due to compressed data formats, among others.

3. Changes in the speaker's acoustic environment: includes background noise, echoes and room acoustics.

Other significant contributors to speaker verification accuracy include within-speaker variations such as mood, health and aging [71], as well as normal recording session variations. This latter contributor is defined as a mismatch between any two recordings of the same speaker, such as the recordings in the classifier training set and those recordings in the testing set [165]. These differ from phonetic variability in that they are not necessarily about the *words* spoken, but instead about tone, speaking speed, and pronunciation, and can also include variations such as uncharacteristic utterances by the speaker. Kinnunen *et al.* cite recording session variability as the "most challenging problem in speaker recognition" [119, p.2]. Bonastre *et al.* [83] state that low error rates for speaker verification are not likely

within the confines of current state of the art, which may help explain the higher than hoped for error rates reported in this chapter.

The DT classification algorithm consumed extensive memory resources when processing such a large dataset, which resulted in many out-of-memory errors during classification. These were solved by moving to a more powerful computer with more available memory and reducing the dataset size to 5000 patterns. Although these strategies supported this research, they were not a viable solution due to resource constraints when this mechanism is deployed in the wild. However, these difficulties must be addressed given that the Transparent Authentication Framework was designed to keep all data on the device in order to protect the device owner's privacy and support a perception of enhanced security. The device owner's data, therefore, cannot be removed from the device to be processed on more powerful computers and the results transmitted back to the device. The chosen pattern classifier must run within the memory and processor constraints of the mobile device.

The memory and processor constraints are not as much of a problem as they appear to be at first glance. The reason for the out-of-memory errors with the voice recordings can be attributed to the volume of data to process rather than the attributes of the pattern classifier. In the working Framework, a much smaller sample of voice data will be used for training, and the testing set in each case will be a single feature vector, or perhaps a small series of features in the case of a longer speaking sequence. Therefore, the out-of-memory errors should not plague the Framework, no matter which classifier is chosen. In the event this does happen, the training and testing sets can be reduced in size to ensure there are no memory issues.

5.7 Speaker Verification in the Transparent Authentication Framework

Speaker verification was considered for inclusion in the Transparent Authentication Framework because voice patterns can be gathered transparently while the mobile device owner completes everyday tasks, such as making phone calls. The results of this study have shown that voice patterns gathered on a mobile device are indeed a viable source of unique information that can be used as part of the Framework to verify the identity of a mobile device owner. These results also indicate that speaker verification is best combined with another biometric in order to improve the error rates revealed in the results of this study. Therefore, it can reasonably be argued that this study has justified the inclusion of speaker verification as a possible biometric in the Transparent Authentication Framework, although it should be acknowledged that a larger study needs to be undertaken to verify these results on a larger sample size prior to including this biometric in a working framework.

5.8 Summary

This chapter has reported on the design, implementation, and results of the speaker verification study that was undertaken to provide support for speaker verification as a useful and meaningful biometric in the Transparent Authentication Framework. The results of the study, although preliminary, suggest that speaker verification may be sufficient to *verify* the identity of a mobile device user, although likely in combination with another biometric. A larger study, as justified by these initial results, is recommended to verify these results and to allow the inclusion of other mobile devices.

Chapter 6

Multimodal Biometrics Feasibility Study

This chapter discusses methods that may be used to fuse the biometrics used in the Transparent Authentication Framework. The fusion of keystroke dynamics and speaker verification into a single multimodal biometric provides a bimodal input for calculating device confidence in the Framework. The two fusion methods, called the Naïve Method and Posterior Probability Method, are presented. The calculations used are detailed, then the data from the previous two studies are tested using these methods and compared to the results from the individual biometrics. The results show that there are improvements in both the error rates for keystroke dynamics and speaker verification reported in the previous two chapters, although the improvements are not statistically significant.

6.1 Study Goals

The purpose of this study is to determine whether combining keystroke dynamics and speaker verification biometrics results in lower error rates when compared to each biometric on its own. This study examines two score-level fusion techniques that are based on the probabilities output by the classifiers. Each technique is tested with the pattern classifiers used in the Keystroke Dynamics (Chapter 4) and Speaker Verification (Chapter 5) studies to determine which performs best in terms of reducing error rates. This study provides an answer to hypothesis H2 of this dissertation, as listed in Section 1.4.1, which asks whether combining biometrics provides a better basis for determining if the owner is the current device user, when compared to using single biometrics. An answer to this hypothesis further informs the overarching goal of this dissertation, which is to provide a framework upon which transparent authentication for mobile devices can be built.

6.2 Fusion Methods

There are two ways to consider biometric fusion [57]. First, a single biometric type (e.g., fingerprint) may be classified using several classifiers, then the individual results combined to create a single probability. The second way, the one that is adopted in this study, is to use two or more *different* biometrics (e.g., keystroke dynamics and speaker verification), classify them individually, and combine the results of these two biometric *modalities* into a single probability. In this latter case, each biometric modality may be presented to a different pattern classifier depending on which one provides the best results for the data presented to it. While this is the approach taken for testing the efficacy of the two methods proposed in this chapter, these methods may also be used with multiple values from the same type of biometric.

Fusion methods for multiple biometric measurements include feature-level, decision-level and score-level fusion [57, 123], as discussed in Chapter 2. Feature-level fusion techniques were excluded because the two biometrics do not have features in common, which means that combining the feature vectors may be more likely to produce higher error rates [57]. Decision-level fusion is also unsuitable because it can result in a multimodal biometric that produces worse error rates than the individual biometrics. This is because it only has access to the decision and not the granularity thereof. Thus, a decision could be made to reject based on a borderline case (i.e., one very close to the threshold), which favors false rejections rather than false positives.

Score-level fusion has been chosen for combining the keystroke dynamics and speaker verification patterns for this research because the two pattern classifiers output a score that is interpreted as a probability that the gathered feature belongs to the device owner. This method does not rely on pattern independence, which is important as some implementations of the Transparent Authentication Framework may use different dependent biometrics.

6.2.1 Score-Level Fusion Techniques

Score-level fusion is a technique in which the scores or probabilities of several biometrics may be combined. The biometric decision has not been made at this point, but the feature vectors have been presented to a classifier that outputs either a probability or a score-match matrix. Many of these methods require normalization to ensure that the different classifier outputs are within the same range.

Score Normalization

The scores to be combined are normalized to ensure that the value they represent comes from the same distribution, say between 0 and 1 [127]. Normalization reduces the effect of differing distributions. For instance, if one classifier outputs a score between 0 and 100 and another classifier outputs a score between 1000 and 2000, the first biometric will have little effect on the fusion result since its scores seem much lower than the second biometric. Without the normalization step, the biometric with the higher range of scores will eliminate the contribution of the biometric with the lower range of scores. Furthermore, score normalization allows the addition or substitution of other biometrics as they become available without considering whether the scores will complement each other.

Common normalization techniques include Min-Max, Z-Score, and TANH, among others [122, 166]. These methods, in general, involve combining the median, maximum and minimum values, and standard deviations of several scores to ensure they fall within the same ranges. The interested reader is directed to the sources cited previously for a detailed discussion of normalization techniques.

Score-Match Matrix Methods

Many studies into multimodal biometrics use well-known score-level fusion methods that are based on a *score-match matrix* [122, 166, 167]. Score-match matrix creation begins with pattern classification. When classifying a particular test feature vector, the new input data is compared to each feature vector in the training set. The comparison results in a distance that represents how different (i.e., how far) the new data is from the training data. For each pair of training and testing data, these distances are put into an n x n score-match matrix, where n is the size of the training set. At the end of this process (i.e., the end of the classifier's testing phase), there is a score-match matrix for every feature vector in the test set.

The score-level fusion techniques that are common in the literature are summarized in Table 6.1. These techniques are presented here to give a sense of the state-of-the-art, and to justify the choice of the Naïve and Posterior Probability Methods for this research. The implementations in Table 6.1 assume that M_i is the score-match matrix from classifier i, and that there are K classifiers in total.

Once all score-match matrices have been created, the fusion methods must normalize and then combine them using the formulae shown in Table 6.1. For instance, the Simple Sum method adds all elements of the matrices together, and the Minimum and Maximum Score methods choose the smallest and largest scores, respectively, resulting in a new score-match matrix that represents all input matrices [168]. These methods imply processing possibly

Name	Formula	Description
Simple Sum	$\sum_{i=1}^{K} M_i$	The score-match matrix values for each biometric are summed to provide a new score-match matrix.
Min Score	$min(M_1, M_2, \ldots, M_K)$	The smallest score for each element in all score-match matrices is selected, creating a combined score-match matrix.
Max Score	$max(M_1, M_2, \ldots, M_K)$	The largest score for each element in all score-match matrices is selected, creating a combined score-match matrix.

Table 6.1: Summary of score-level biometric fusion methods that use score-match matrices.

large and complex data structures. This represents a potentially significant amount of processor and memory use, both of which are constrained on mobile devices. For these reasons, fusion methods that rely on score-match matrices are discounted from use in this research. Instead, fusion methods that take probabilities as input were used rather than those that use score-match matrices.

Sum and Product of Probabilities

Two common score-level fusion techniques that use output probabilities, Sum of Probabilities and Product of Probabilities, were considered for this study [166, 168]. These methods, summarized in Table 6.2, calculate the sum and product of the posterior probabilities of a class given the input data. While these methods seem ideal for this work, the literature does not provide adequate information to recreate such methods. When using the sum method, for instance, there is no discussion of normalization techniques except to state that they are "implied in the algorithm" [166]. Due to the lack of implementation details, these methods were discounted from use in this study.

Name	Formula	Description
Sum of Probabilities	$\sum_{i=1}^{K} P(Owner \mid M_i)$	Probabilities for all biometrics are summed to create a single, combined probability.
Product of Probabilities	$\prod_{i=1}^{K} P(Owner \mid M_i)$	Probabilities for all biometrics are multiplied to create a single, combined probability.

Table 6.2: Summary of probability-based score-level biometric fusion methods.

6.2.2 Sequential Probability Ratio Test

Another option for combining biometrics in the Transparent Authentication Framework is using the Sequential Probability Ratio Test (SPRT). With this test, the Framework would continue to gather biometric samples until a confident decision within two thresholds can be made. Once the threshold is met, the biometric probabilities and decisions can be stored until needed to raise the probability further. This method lends itself well to use in the Transparent Authentication Framework, although it was not tested specifically in this research because it may prove too slow in adjusting the device confidence in the presence of an attacker. This assertion may be tested in a simulated implementation of the Framework, and is thus left for future work.

Research has been performed that uses SPRT with multimodal biometrics, with good results [169].

Two score-level probability fusion methods have been selected for testing with the keystroke dynamics and speaker verification data: the Naïve and Posterior Probability Methods, as discussed in the next section.

6.3 Combining Biometrics for Score-Level Fusion

This study uses the results from the keystroke dynamics and speaker verification experiments first to create a multimodal biometric, then to compare the error rates for multimodal biometric to those of the individual biometrics. The combination methods presented in the following two sections, the Naïve Method and the Posterior Probability Method, fuse the probabilities output by the two classifiers to calculate a new probability. This new probability is then used to make a decision regarding whether both patterns belong to the device owner. Both combination methods take advantage of the mathematical rules of probabilities.

An important consideration when combining probabilities is whether the samples are independent. In probability theory, stating that two events are independent means that the observed value of one probability does not affect the other. Keystroke dynamics and speaker verification biometrics are not truly independent since they are gathered from the same person. However, the likelihood that they are completely independent or completely dependent on each other is low. In this case, because the two values are independent measurements (and, in fact, measure different characteristics of the same person), the values are considered *conditionally independent* given the class (either owner or rest-of-world).

This explains independence in the case of the owner keystroke and voice patterns. In the case of the rest-of-world patterns, there are some cases where the keystrokes of one owner are matched with the voice of another since the rest-of-world patterns are not matched by

owner before combination. For example, a posterior probability from Owner A's keystroke pattern may be combined with a conditional probability from a voice sample from Owner B. Since both samples are not from the designated device owner, they both have the known class of rest-of-world. Combining biometrics in this manner is likely more stringent than ensuring that both probabilities originate with the same person, although this is not proven in this research. In this case, true independence is assumed because the two measurements are independent and also come from two different people.

Symbol	Meaning
C_O	Class decision for owner
C_W	Class decision for world (i.e., not owner)
D_i	Data from an unspecified biometric or explicit authentication attempt
D_{KD}	The keystroke dynamics data used for classification (i.e., the feature vector)
D_{SV}	The speaker verification data used for classification (i.e., the feature vector)
$P(C_O\|D_{KD})$	Conditional posterior probability of Owner class given the keystroke dynamics data (i.e., the classifier output)
$P(C_O\|D_{SV})$	Conditional posterior probability of Owner class given the speaker verification data (i.e., the classifier output)
$P(C_O\|D_{KD}, D_{SV})$	The combined probability of the Owner class given the keystroke dynamics and speaker verification data
$P(C_W\|D_{KD}, D_{SV})$	The combined probability of the World class given the keystroke dynamics and speaker verification data

Table 6.3: Definitions and notation of terms for biometric combination methods.

6.3.1 Naïve Method

The first method by which biometrics have been combined is a naïve approach based on the mathematical rules for probabilities. With this method, conditional independence between the two values being combined is assumed. This approach, which here is called the Naïve approach since it is a simple way to consider biometric combination, is calculated by subtracting the product of probabilities from their sum:

$$P(C_O \mid D_{KD}, D_{SV}) = [P(C_O \mid D_{KD}) + P(C_O \mid D_{SV})] - [P(C_O \mid D_{KD}) * P(C_O \mid D_{SV})] \tag{6.1}$$

This method, while unsuitable for mutually exclusive events, allows for some overlap between the the two classes, as shown in Figure 6.1. However, when two probabilities are added, the intersection is added twice, which leads to error in the reported combination. The intersection is removed by subtracting the intersection once by subtracting the product of the two probabilities.

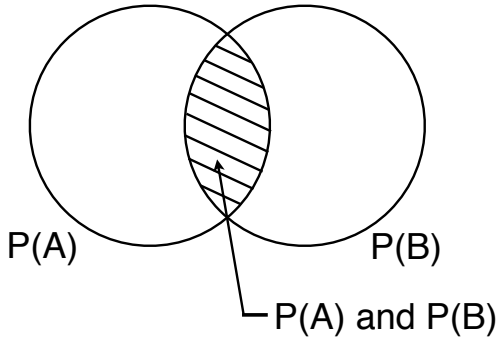


Figure 6.1: Overlap between two probabilities. When adding $P(A)+P(B)$, the overlap area is added twice. It is removed by subtracting the intersection, $P(A) * P(B)$.

The Naïve approach does not require an explicit normalization step; it is implied in Equation (6.1). This equation can also be written for more than two biometrics. This is helpful since the Transparent Authentication Framework may use more than two:

$$P(C_O \mid D_1, D_2, \ldots D_K) = [\sum_{i=1}^{K} P(C_O \mid D_i)] - [\prod_{i=1}^{K} P(C_O \mid D_i)] \tag{6.2}$$

6.3.2 Posterior Probability Method

The second combination method takes into account the designated class given the input data. This approach, called the Posterior Probability Method here, was extended from the method for combining the output of two or more classifiers described in [170]. Its derivation depends on Bayes' Rule, which can be used to relate the probabilities of two events (A and B) before and after conditioning on a third event (C). In terms of this research, the two events A and B are the determination that the biometric sample belongs to the device owner given the keystroke dynamics and speaker verification feature vectors, respectively. The third event, C, is the determination of either owner or rest-of-world class. Bayes' rule is represented as follows, using the notation given in Table 6.3:

$$P(C_O \mid D_{KD}, D_{SV}) = \frac{P(D_{KD}, D_{SV} \mid C_O) \cdot P(C_O)}{P(D_{KD}, D_{SV})} \tag{6.3}$$

where it is assumed that D_{KD} and D_{SV} are conditionally independent measures given the class C_O:

$$P(D_{KD}, D_{SV} \mid C_O) = P(D_{KD} \mid C_O) \cdot P(D_{SV} \mid C_O) \tag{6.4}$$

Substituting Equation (6.4) into Equation (6.3) results in:

$$P(C_O \mid D_{KD}, D_{SV}) = \frac{P(D_{KD} \mid C_O) \cdot P(D_{SV} \mid C_O) \cdot P(C_O)}{P(D_{KD}, D_{SV})} \tag{6.5}$$

The next equation is the formula used for the Posterior Probability Method. The normalization constant, α, ensures that the probabilities of the possible classes sum to 1:

$$P(C_O \mid D_{KD}, D_{SV}) = \alpha \cdot \frac{P(C_O \mid D_{KD}) \cdot P(C_O \mid D_{SV})}{P(C_O)} \tag{6.6}$$

The derivation of α depends on the constraint that the probabilities of the two classes C_O and C_W must sum to 1 for the same input data, either D_{KD} or D_{SV}. This constraint is written as:

$$P(C_O \mid D_{KD}, D_{SV}) + P(C_W \mid D_{KD}, D_{SV}) = 1 \tag{6.7}$$

Thus, the derivation of α begins with the above assumption for the two classes in this work, C_O and C_W:

$$1 = \alpha \cdot \left[\frac{P(C_O \mid D_{KD}) \cdot P(C_O \mid D_{SV})}{P(C_O)} + \frac{P(C_W \mid D_{KD}) \cdot P(C_W \mid D_{SV})}{P(C_W)} \right] \tag{6.8}$$

Rearranging trivially to move alpha to the left hand side gives:

$$\frac{1}{\alpha} = \left[\frac{P(C_O \mid D_{KD}) \cdot P(C_O \mid D_{SV})}{P(C_O)} + \frac{P(C_W \mid D_{KD}) \cdot P(C_W \mid D_{SV})}{P(C_W)} \right] \tag{6.9}$$

Note that $P(C_W) = 1 - P(C_O)$ since there are only two possible classes. Substituting this into Equation (6.9) gives:

$$\frac{1}{\alpha} = \frac{P(C_O \mid D_{KD}) \cdot P(C_O \mid D_{SV})}{P(C_O)} + \frac{(1 - P(C_O \mid D_{KD})) \cdot (1 - P(C_O \mid D_{SV}))}{(1 - P(C_O))} \tag{6.10}$$

For simplicity (as is done in [170]) assume $X = P(C_O \mid D_{KD})$, $Y = P(C_O \mid D_{SV})$ and $Z = P(C_O)$ and substitute into Equation (6.10):

$$\frac{1}{\alpha} = \frac{XY}{Z} + \frac{(1-X)\cdot(1-Y)}{(1-Z)} \tag{6.11}$$

Simplifying by creating a common denominator gives:

$$\frac{1}{\alpha} = \frac{XY - XYZ + Z(1-Y-X+XY)}{Z\cdot(1-Z)} \tag{6.12}$$

$$= \frac{XY - XYZ + Z - YZ - XZ + XYZ}{Z\cdot(1-Z)} \tag{6.13}$$

$$= \frac{Z\cdot(1-Y-X) + XY}{Z\cdot(1-Z)} \tag{6.14}$$

Substituting into Equation (6.6) (replacing values with X, Y and Z where necessary) gives:

$$P(C_O \mid D_{KD}, D_{SV}) = \frac{Z\cdot(1-Z)}{Z\cdot(1-Y-X)+XY} \cdot \frac{XY}{Z} \tag{6.15}$$

$$= \frac{Z\cdot(1-Z)\cdot XY}{Z\cdot(Z\cdot(1-Y-X)+XY} \tag{6.16}$$

$$= \frac{XY\cdot(1-Z)}{Z\cdot(1-Y-X)+XY} \tag{6.17}$$

Replacing the original values of X, Y and Z into the final equation above gives the final equation for the Posterior Probability Method of biometric combination:

$$P(C_O \mid D_{KD}, D_{SV}) = \frac{P(C_O \mid D_{KD})\cdot P(C_O \mid D_{SV})\cdot(1-P(C_O))}{P(C_O)\cdot(1-P(C_O \mid D_{SV})-P(C_O \mid D_{KD}))+(P(C_O \mid D_{KD})\cdot P(C_O \mid D_{SV}))} \tag{6.18}$$

Since the Transparent Authentication Framework is intended for use with any number of biometrics rather than just two, it is helpful to rewrite Equation (6.18) for several inputs, as follows:

$$P(C_O \mid D_1, \ldots, D_K) = \alpha \cdot \frac{\prod_{i=1}^{n=K} P(C_O \mid D_n)}{P(C_O)^{K-1}} \tag{6.19}$$

The α term is left out for brevity, and because its exact calculation depends on the number of terms (i.e., biometrics) used.

6.4 Study Design

The independent variable for this study is the combination method used: Keystroke Dynamics (KD), Speaker Verification (SV), Naïve Method (NM) or Posterior Probability Method (PPM). While the first two are not combination methods, they are included here for comparison purposes in order to determine whether either of the latter two methods show improvements over the two individual biometrics.

6.4.1 Participants

The biometric data used in this study was from the Keystroke Dynamics and Speaker Verification studies described in Chapters 4 and 5, respectively. The participants for this study are those who participated in both the keystroke and speaker studies; the overlap yielded 6 participants. They ranged in age from early 20s to late 50s, and had a range of experience with both typing and speaking on their device.

The keystroke and voice patterns used in the multimodal biometric come from the same owner. In some multimodal biometric studies, biometrics coming from two different owners are combined together to create an imaginary individual in order to determine whether this also increases or decreases the error rates when compared to the individual biometrics. However, this implies that there are known biometric samples from someone other than the device owner. This is not known on a mobile device, where it is assumed there is a single owner, and any biometric decisions are simply likelihoods, not certainties, that the pattern belongs to the device owner.

6.4.2 Apparatus and Materials

The two studies that provided the data for this experiment used iPhones and iPod Touches to gather the data, as detailed in Chapters 4 and 5. The only other equipment required for this study was MatLab version R2012b, which was used to perform the NM and PPM calculations described in Section 6.3 and to perform the statistical significance tests presented in Section 6.6.

6.4.3 Procedure

The reported results from the Keystroke Dynamics and Speaker Verification studies are cross-validation averages, meaning that the data is segmented into ten training and testing sets and classified individually. This usually provides a stronger sense of the type of data

reported, and to validate the results of a single data run. To this end, there were no single posterior probabilities for each pattern and for each classifier to use as input for this study. Instead, the exact method (short of cross-validation efforts) described in Chapters 4 and 5 was used to execute a single data run for each classifier, using the same data as in the two original studies, and ensuring that the same data was presented to each classifier. The output posterior probabilities were then used as described here, and are referred to in the results tables as keystroke dynamics (KD) and speaker verification (SV) biometric classification.

The keystroke dynamics and speaker verification classifications included posterior probabilities that a given pattern belonged to the device owner. These probabilities were calculated for the five different pattern classifiers. For each of the six owners in this study, the posterior probabilities from the keystroke dynamics data were matched with posterior probabilities from the speaker verification study, ensuring that both original biometric samples were from the same owner. Since there was far more speaker verification than keystroke dynamics data, only enough samples from the former were used to match one to each keystroke vector. As seen in Figure 6.2, the keystroke and voice data are then presented to classifiers that output a probability that the input data belongs to the owner. Next, the posterior probabilities were combined pairwise using the NM and PPM methods to produce combined probabilities. These were then converted to a biometric decision, D, using the following rule:

$$D = \begin{cases} 0 & \text{if } P(C_O \mid D_{KD}, D_{SV}) < 0.5 \\ 1 & \text{if } P(C_O \mid D_{KD}, D_{SV}) \geq 0.5 \end{cases}$$

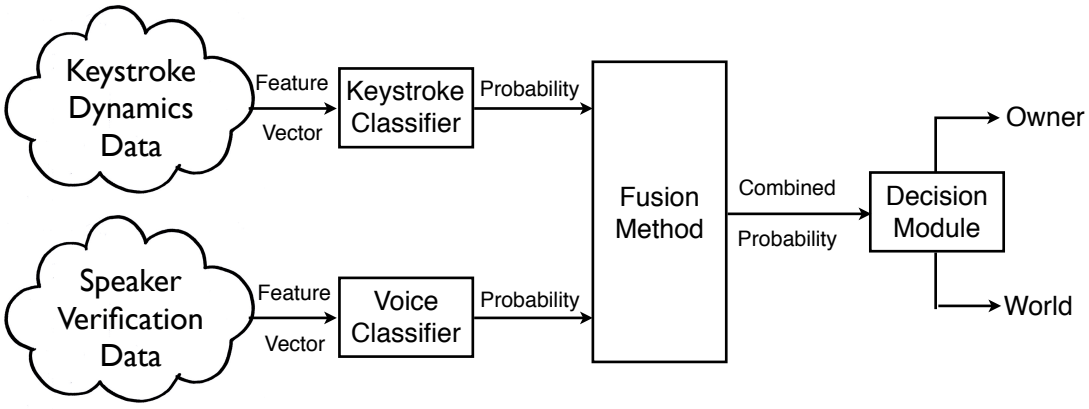


Figure 6.2: Procedure for multimodal biometric fusion.

The new decisions were then compared to the known decisions, which were created using the known classes of each biometric test set. These known classes are vectors of 0 and 1 values that represent whether the given test feature vector belonged to the owner or not, where 1 represented an owner pattern. The comparison process is shown in Figure 6.3. The new

known classes were created by ORing the known class vectors for keystroke dynamics and speaker verification test sets. Since the same number of keystroke dynamics patterns from the owner were paired with owner patterns from speaker verification and the same for world patterns, ORing the two known class vectors resulted in only two calculations: 1 OR $1 = 1$, 0 OR $0 = 0$. In no cases was an owner pattern paired with a world pattern, meaning that the calculations 1 OR $0 = 1$, 0 OR $1 = 1$ were never performed. This improves the classification rate in cases where there may be a disagreement between the component biometrics, even if they both belong to the owner (i.e., one of the classifiers was incorrect).

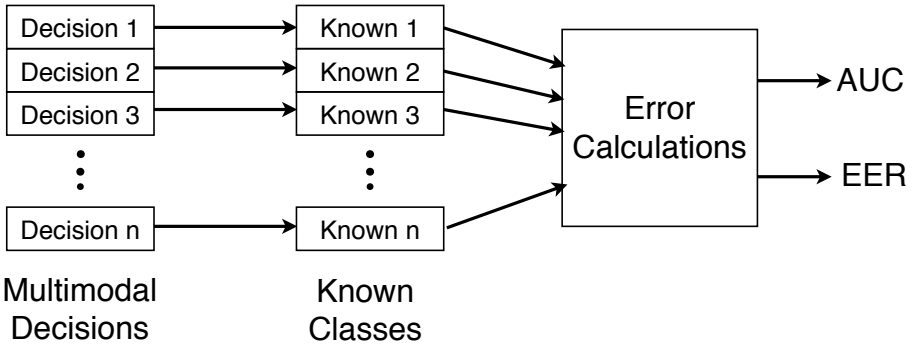


Figure 6.3: Comparison of multimodal decisions to known classes.

Once this new vector of known decisions was calculated, it was used with the newly calculated combined probabilities to calculate an ROC curve from which the AUC and EER values may be calculated. Finally, the new AUC and EER values were compared to those produced by the individual biometric classifiers. The calculations for pattern classification, EER, ROC curves and associated AUC values were performed in MatLab version R2012b.

6.4.4 Biometric Weighting

In the Transparent Authentication Framework, each biometric can be weighted according to how likely it is to represent the owner. For instance, in the case where an owner types frequently but does not make many phone calls, there would be far more keystroke than voice patterns. This affects the classifiers' ability to classify data correctly since the speaker verification classifier would have trained on far less data and therefore would be more prone to errors. In this case, the keystroke dynamics biometrics could be assigned a higher weight than the speaker verification biometrics so that the latter does not negatively affect device confidence calculations unnecessarily.

Such weighting provides flexibility to the Framework, but the weighting is intended to take place before the biometric combination method is used. The result of the weighting would be a lower or higher probability than the original. However, this chapter is intended to examine

the actual combination methods, irrespective of weighting, which is considered part of the Framework. In essence, the weighting itself does not affect the validity of these results since the probabilities used are from a range of possible values, and could represent either weighted or unweighted probabilities. Furthermore, testing a weighting scheme is dependent on the particular application that is based on the Framework, since it has a direct effect on the provision of security. For these reasons, testing a weighting scheme is best left for a simulation phase, which is future work.

6.5 Pattern Classification

The classifiers used in this study were the same as for the Keystroke Dynamics and Speaker Verification studies: k-Nearest Neighbor with Manhattan (k-NN (Man)) and Euclidean (k-NN (Eucl)) distance measures, Decision Tree (DT), and Naïve Bayes with both Kernel Density (NB(KD)) and Gaussian (NB(Gau)) distributions. In order to simplify the process, the two biometrics combined for this study came from the same owner and from the same classifier, although in practice these decisions may come from different classifiers without affecting the method or results. Classifier variations are not considered significant in the context of this discussion because each classifier outputs a probability that the input pattern came from the device owner. These probabilities range between 0 and 1 no matter which classifier is used and are combined in the same way. If the classifier output is a low probability, it will affect the multimodal biometric in the same way, no matter which classifier it came from, and similarly for a high probability. Moreover, all classifiers may output unusually high or low probabilities due to errors, even in the presence of a usually easily classified pattern. The feasibility studies conducted for the two biometrics were undertaken to identify a "best" classifier, which should be used for all future biometric choices.

6.6 Results and Analysis

This section reports the results and analysis of the Multimodal Biometrics study. The EER and AUC values for each test are reported, but FAR and FRR values are not. In a single-biometric situation, FAR and FRR are easily calculated from where the classifier output was different than the known class of the test case.

Multimodal biometrics test cases from six participants were considered in the study. They were compared to the known classes of the test cases and used to calculate the EER and AUC values shown in Tables 6.4 and 6.5 respectively.

Table 6.4 shows the EER results by owner for the two single biometrics, KD and SV, as

Owner	Metric	Classifier				
		Naïve Bayes		Decision	5–NN	
		Gaussian	Kernel Density	Tree	Euclidean	Manhattan
Owner1	KD	47.89	27.27	30.08	47.39	38.76
	SV	39.44	45.07	42.22	36.36	36.03
	NM	45.07	27.27	34.72	44.65	36.56
	PPM	49.30	36.62	35.95	34.97	35.27
Owner2	KD	31.88	26.09	37.10	28.64	27.21
	SV	46.15	38.46	49.57	38.36	36.95
	NM	38.46	28.99	41.95	32.55	25.49
	PPM	31.88	26.09	42.63	28.64	21.39
Owner3	KD	45.00	33.87	22.30	22.53	14.93
	SV	40.00	40.00	49.10	39.90	41.93
	NM	41.94	30.00	35.73	25.49	20.00
	PPM	40.00	30.00	25.41	28.22	19.72
Owner4	KD	50.00	16.67	25.76	32.76	29.04
	SV	40.79	42.11	55.47	53.98	50.33
	NM	47.37	16.67	50.36	47.50	49.35
	PPM	50.00	33.33	30.04	44.44	37.77
Owner5	KD	40.00	30.00	43.55	21.68	20.60
	SV	36.11	44.44	41.46	29.51	20.00
	NM	40.00	30.00	47.07	19.70	20.00
	PPM	40.00	30.00	32.35	18.35	16.67
Owner6	KD	33.33	11.11	11.09	34.63	28.37
	SV	44.44	39.73	43.58	45.80	40.84
	NM	38.36	11.11	20.69	42.44	37.71
	PPM	44.44	8.22	11.11	39.24	33.55

Table 6.4: EER values (%) for Keystroke Dynamics (KD), Speaker Verification (SV), Naïve Method (NM) and Posterior Probability Method (PPM).

well as for the two combination methods, NM and PPM. For each owner, the NM and PPM were generally better than KD and SV in that they produced somewhat lower EER values. For example, for Owner5 the EER was between 20.60% and 43.55% for KD and between 20.00% and 44.44% for SV, which means that the classifier was better than chance for both biometrics, although not by much in most cases. When using the NM combination method, the EER was between 19.70% and 47.07%, which is a nominal improvement. For PPM, Owner5's results were also slightly improved over individual methods.

Note that the NM and PPM values do not represent the combination of the KD and SV EER values. They are calculated as combinations of the individual biometric posterior probabilities and compared to a different set of known classes. Therefore, the SP and PPM EER

values are not expected to differ from the KD and SV EER values in a regular manner.

The individual and combination AUC values are presented in Table 6.5, and have better results when compared to the EER values. For instance, Owner5's AUC values range from a low of 60.07% to a high of 85.83% for KD, with a similar range for SV. The NM method showed little improvement over the individual biometrics, likely due to the limitations in that calculation. The PPM combination method showed improvement over the individual methods, with a low of 65.14% and a high of 90.00%.

Owner	Metric	Classifier				
		Naïve Bayes		Decision Tree	5–NN	
		Gaussian	Kernel Density		Euclidean	Manhattan
Owner1	KD	59.15	79.64	76.95	50.45	63.76
	SV	62.48	59.41	61.84	69.65	68.31
	NM	59.92	79.45	72.61	58.77	67.86
	PPM	58.45	67.35	70.42	67.16	71.06
Owner2	KD	70.79	84.17	68.73	79.15	82.39
	SV	62.10	57.97	51.62	66.22	66.50
	NM	73.13	79.15	63.44	74.75	78.14
	PPM	74.69	82.61	62.88	81.10	86.90
Owner3	KD	60.89	78.31	83.15	84.07	88.27
	SV	64.68	64.35	53.27	61.73	61.45
	NM	63.31	79.60	72.21	82.10	86.08
	PPM	61.21	80.16	78.06	80.73	85.89
Owner4	KD	55.70	90.79	82.13	70.18	78.40
	SV	62.50	65.57	40.13	42.98	50.77
	NM	63.38	92.54	49.28	54.71	52.30
	PPM	59.43	82.89	75.11	57.13	72.04
Owner5	KD	63.19	78.75	60.07	80.97	85.83
	SV	69.86	50.14	63.12	80.69	78.19
	NM	61.67	78.61	55.54	87.22	84.24
	PPM	65.14	69.17	69.72	90.00	91.11
Owner6	KD	64.84	97.41	93.15	66.29	71.08
	SV	65.91	60.12	59.51	57.76	63.01
	NM	70.78	95.74	86.28	62.40	67.81
	PPM	66.97	97.87	92.16	66.74	70.70

Table 6.5: AUC values (%) for Keystroke Dynamics (KD), Speaker Verification (SV), Naïve Method (NM) and Posterior Probability Method (PPM).

For both EER and AUC calculations, NM tended to be somewhat more optimistic since the calculation itself takes fewer factors into account (i.e., the constraint relationship between owner and world class probabilities). This optimism means that for a combination that is

close to the threshold separating owner from world, using this calculation may output the Owner class more often that the World class. Using PPM, on the other hand, resulted in stricter class determinations, as shown by the slightly lower AUC value than NM in most cases. This strictness has the opposite effect on close decisions; using this calculation method is more likely to output the World class.

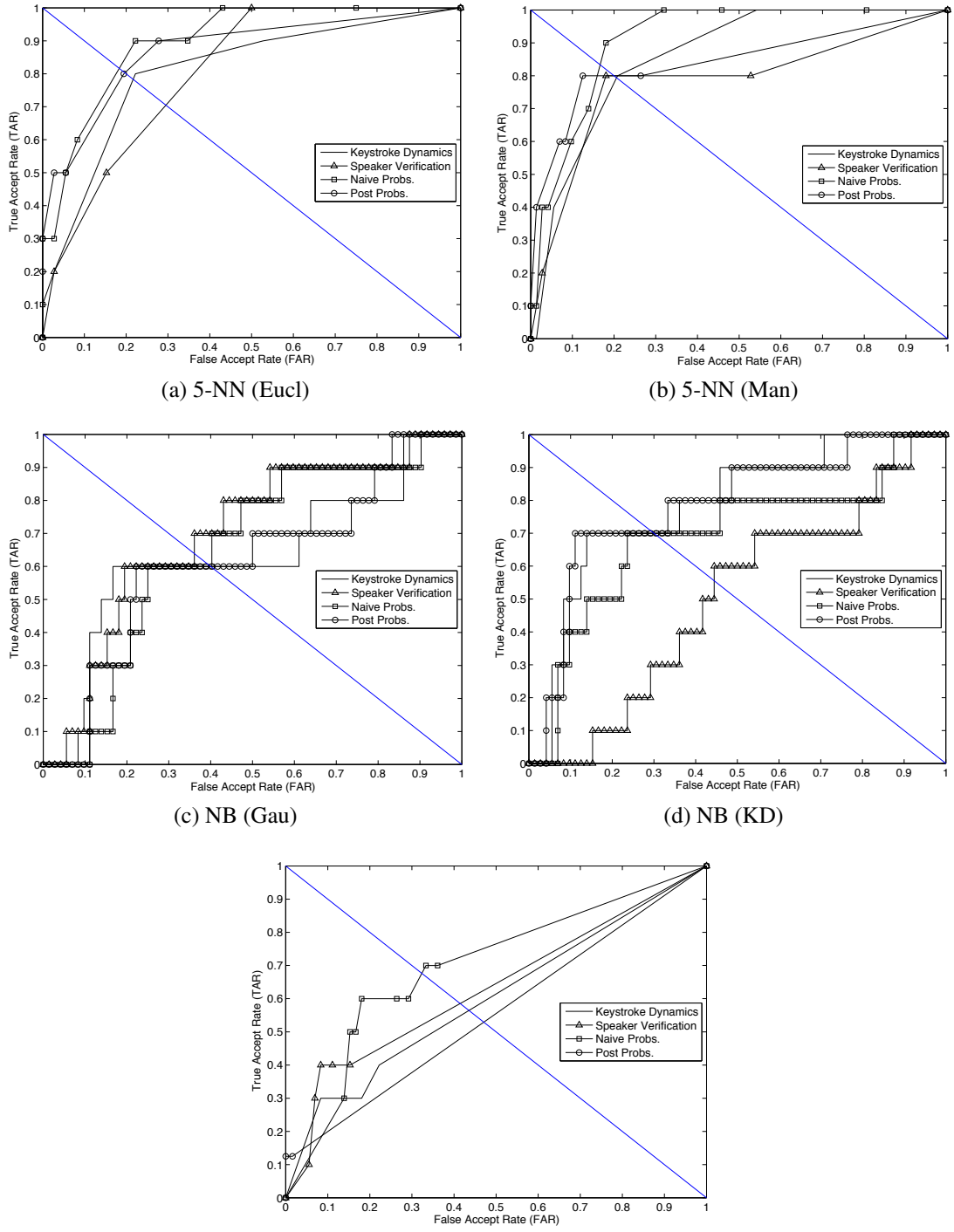
Neither of these issues supports the selection of one combination method over the other. Instead, since NM is optimistic and thus may allow more false positives, this method should be used in cases where lower security levels are acceptable since a false positive represents a potential breach. PPM's stricter decision-making means that there may be a higher level of false negatives, which may annoy owners in low security situations, since they are being unnecessarily prevented from accessing the protected services. Therefore, PPM is better used in situations that require higher security, where the higher false negative occurrence is an acceptable tradeoff for an increased security level.

One of the reasons that PPM represents a stricter decision process has to do with the KD and SV posterior probabilities that are input into the equation. In many cases, the classifier was unable to make a decision regarding the class to which a particular feature vector should belong. In these cases, the posterior probability for owner ($P(C_O \mid D_{KD})$ or $P(C_O \mid D_{SV})$) was zero, as is also the case when the feature vector was determined to not belong to the owner. If a zero value is substituted into Equation (**??**), the numerator is also zero, which means that the combination of the two posterior probabilities will be zero. This is particularly concerning in situations where one biometric has a very high probability output, and the other has zero probability. It is expected that this situation will happen with less frequency if there is more data with which to train the classifiers.

The improvements shown with the two combination methods must be considered carefully for several reasons. First, this is a very small dataset – much smaller than those from the keystroke dynamics and speaker verification studies, and therefore is subject to the same issues. Specifically, the classifiers are trained on a small amount of data and may not accurately model the owner's patterns, and tested on an even smaller set, which gives very little data upon which to base EER and AUC calculations. This in turn means that individual misclassifications represent a much larger portion of the total number of classifications. EER in particular may be artificially high in the individual classifiers. For these reasons, the combination methods may present a positively biased case, particularly NM.

The ROC curves for each combination method are represented pictorially in Figure 6.4. This figure shows the results for Owner5 only; the results for the other owners are similar and have been therefore left out for brevity.

These ROC curve comparisons that can be made further enhance the differences between the NM and PPM methods. The figures show that NM and PPM are only better than the



(a) 5-NN (Eucl)

(b) 5-NN (Man)

(c) NB (Gau)

(d) NB (KD)

(e) DT

Figure 6.4: ROC curves for Owner5 over all classifiers. The point where the diagonal line crosses each curve is the EER for that curve.

individual biometrics up to a certain FAR value in some cases. For instance, in Figure 6.4c the AUC values increase for all situations in a similar step-wise manner. This shows that there is little improvement in combining classifier output for Naïve Bayes Gaussian, and that at some thresholds (e.g., up to 0.2 on the x-axis) KD is actually the better choice in terms of

a higher AUC. Such a result can be explained by lack of input data or a poor match between classifier and data.

In Figure 6.4a, the AUC values for NM and PPM are considerably higher than for KD and SV at most thresholds. This shows that in some situations, combining biometrics provides better data upon which to base a decision of owner or world. This may be explained by the classifier being better matched to the data provided. The results of the keystroke dynamics and speaker verification studies support the conclusion that some classifiers produce better output depending on the type of data presented to them.

The figures representing the DT and both 5-NN classifiers seem to have far fewer datapoints than the two Naïve Bayes figures. In actuality, all five figures have the same number of datapoints since they were generated from the results of presenting exactly the same data to all five classifiers. However, many of the posterior probabilities for the 5-NN classification were either exactly 0% or 100%. The reason for this lies in the method by which posterior probabilities are calculated for k-NN classifiers. With k-NN classifiers, all training data is plotted on an n-dimensional space, where n is the number of features. These training values also have a known classification associated with them. Test data is also plotted on the same space, and the determination for its class is made by polling the k nearest neighbors and choosing the most common value[1]. The posterior probability is then calculated by averaging the known classes of the k nearest neighbors. For instance, if the 5 nearest neighbors in this example all had a known class of 1, then the posterior probability would be $\frac{1+1+1+1+1}{5} = \frac{5}{5} = 1$. The calculation is similar if the 5 nearest neighbors all had a known class of 0. However, if the 5 nearest neighbors had different known classes (say 0,1,1,0, and 1), then the posterior probability would be calculated as $\frac{0+1+1+0+1}{5} = \frac{3}{5} = 0.6 = 60\%$. Therefore, it is not uncommon to see a large number of 0% and 100% posterior probabilities with k-NN classifiers.

The results in all cases show that the NM and PPM methods hold promise, although this result should be considered as a preliminary finding, and confirmed with a larger study that collects more data from participants for analysis. The selection of which method to use is left to the developer, since the risk profile of the application under development must be used to make this decision.

6.6.1 Statistical Significance

As with the Keystroke Dynamics and Speaker Verification feasibility studies, tests of statistical significance were used to determine whether differences in reported error rates were

[1]There are other methods of selecting the value other than most common, but this was the one chosen for this implementation.

due to the different combination methods or attributable to random effects. Similar to these studies, the values being compared are EER and AUC, which are non-parametric as shown by the results of the Kolmogorov-Smirnov test of distribution shape (Table 6.6).

Metric	KD	SV	NM	PPM
EER	< 0.0001	< 0.0001	< 0.0001	< 0.0001
AUC	< 0.0001	< 0.0001	< 0.0001	< 0.0001

Table 6.6: Results of the Kolmogorov-Smirnov test of distribution shape for EER and AUC values. Since each value is less than the $\alpha = 0.05$ significance level, then each distribution of data is considered significantly different from a normal distribution (i.e., the distribution for these values is non–parametric).

In addition to having a non-parametric distribution, the multimodal biometrics data has been compared in a pairwise manner (i.e., comparisons between two groups only). For these reasons, the Wilcoxon Signed-Rank test was used to determine whether the AUC and EER values from the multimodal biometrics test were statistically significantly different.

As discussed in the previous section, using the two combination methods resulted in higher AUC values and somewhat lower EER values than using keystroke dynamics or speaker verification results separately. The values in Table 6.7 were calculated first by selecting a biometric method (KD, SV, NM or PPM) and taking all values in the corresponding row for all owners and all classifiers and computing their median. Then, the statistical significance for the medians was compared in a pairwise manner. These results are the bracketed values shown below each median. In all cases, the KD EER values were considered statistically significantly better than SV and NM, but not the PPM ($\alpha = 0.05$). The PPM and NM methods were not considered significantly different, which means that the selection of the KD EER median as "best" because it was the lowest value does not prove it truly was best when compared to the other results. Similarly, the AUC KD EER median value was chosen as "best" because it was the highest value, although it was only significantly better than SV. As with the EER results, the lack of statistical significance means no determination of the "best" method can be made with these results. For both the EER and AUC values, the differences in results can be attributed to chance. This indicates that a study with more data and perhaps more participants is justified.

These statistical significance test results, while important, should be considered with care. Such tests are strongly affected by the number of participants and the amount of data used in their calculation. In this study, there were six participants, and the same data samples were used for each test. While this study size is acceptable in conducting a feasibility study such as this one, it is unlikely to represent the target population with enough accuracy to allow a statistical significance test to deliver meaningful results. They are presented here for

Metric	KD	SV	NM	PPM
EER (%)	**29.52** –	41.15 (< 0.0001)	36.15 (0.0082)	32.84 (0.1986)
AUC (%)	**78.36** –	62.29 (< 0.0001)	72.41 (0.2623)	71.55 (0.1254)

Table 6.7: EER and AUC medians for all classifiers and all owners. The bolded numbers represent the best classifier based on either EER or AUC, and the number in brackets after each percentage is the result of the Wilcoxon Signed-Rank Test, ρ.

completeness and to support the use of both combination methods in a larger study.

6.7 Multimodal Biometrics in the Transparent Authentication Framework

Keystroke dynamics and speaker verification, while promising, have not proven sufficiently distinctive to justify their use as the *sole* decision maker for supporting authentication. To overcome the higher error rates seen in the two biometric studies, a multimodal biometric can be used to reduce the overall error rates to a more acceptable level. This study has shown that the error rates for a multimodal biometric consisting of keystroke dynamics and speaker verification biometrics result in a lower overall error rate. This provides a strong indication that using a multimodal biometric could well enhance the authentication provided in the Transparent Authentication Framework. However, individual biometrics may still be used in the case where a device owner provides only a single biometric modality.

6.7.1 Limitations of the Study

Possible study limitations include the following:

Small Study Size: As with the other studies in this dissertation, the results presented are from a feasibility study that is small both in terms of number of participants and amount of data. This has a strong effect on the reported error rates and the determination of statistical significance.

Unbalanced Datasets: Again, as with the other studies in this work, the datasets were unbalanced in terms of number of owner and rest–of–world patterns. This also has an effect on the FAR and FRR (and thus EER) rates.

Same Classifier: The posterior probabilities for each combined pattern came from the same classifier, although it is possible to combine posterior probabilities of different classifiers. It is not expected that the source of the probabilities will make a significant difference, but tests should be performed to examine this.

6.8 Summary

This chapter has detailed the study designed to test the viability of a multimodal biometric using the classifier output from the Keystroke Dynamics and Speaker Verification studies. This study compared two probability-based score-level fusion techniques to determine which provided the lowest overall error rates. Combining the multimodal biometrics using two score-level fusion techniques resulted in an overall reduction in EER, and AUC increase, although these differences were not statistically significant and thus may be attributed to chance. These results support other research that has concluded that multimodal biometric combination is a viable method for achieving lower error rates, and justify the inclusion of multimodal biometrics in the Transparent Authentication Framework.

Chapter 7

Transparent Authentication Perceptions Study

This chapter describes the Transparent Authentication Perceptions Study[1], which was performed in order to gain perspective regarding users' willingness to use, and opinions on the design of, a transparent authentication mechanism on a mobile device. This chapter provides the study design details, including details of participants, materials, and methodology as well as reports on the study findings. A discussion of the impact of the study's results and its role in the framework presented as part of this research rounds out the chapter.

7.1 Study Goals

Alternative authentication methods have been widely researched over the last decade, but rarely deployed outside a lab setting. The reasons for this vary depending on the features provided by such systems, but may be attributed to lack of user knowledge or a misunderstanding of user wants and needs. The consequence is that researchers do not fully understand how users will use, bypass or accept new security mechanisms. It is therefore beneficial to determine during the design of such systems whether users would be willing to use the system and what functionality they find important. The feasibility studies reported in Chapters 4, 5 and 6 have shown that behavioral biometrics, particularly multimodal, show promise as the basis for the decision–making in a transparent authentication system. The outstanding question is whether mobile device users would choose to use such a method to protect their devices and data. The first purpose of this study is thus to determine whether the participants feel a transparent authentication method on a mobile device provides adequate security, and whether they would consider using it on their own mobile devices.

[1]University of Glasgow ethics approval number CSE01076

The second purpose of this study is to employ user opinions and suggestions to inform the design of the Transparent Authentication Framework presented in this dissertation. Including user-requested functionality in the final Framework ensures that it is not simply a product of a research-focused endeavor. The findings of this study provide justification for further research into transparent authentication for mobile devices. In this way the user, an important stakeholder, has been consulted and their suggestions considered in the design phase of the Framework.

This study attempts to answer several research questions that are related to the study's goals:

1. What are the participant's opinions of, and reactions to, using a transparent authentication method on a mobile device?
2. What is the participant's perceived level of security while using a mobile device that employs transparent authentication?
3. Do the participants find the transparent authentication method easy or difficult to use?
4. Do participants find transparent authentication generally helpful or mostly a hindrance?
5. Would participants choose to use a transparent authentication method on their own mobile device, if one were available?
6. How do participants react to barriers blocking them from completing their intended tasks, in terms of frequency?

To determine the answers to these questions, an iPhone application was developed that presented participants with a series of tasks to complete. Such a study would normally require a fully-functional transparent authentication method on the device. Such a method does not yet exist, thus the study was designed as a Wizard of Oz study [171], in which the participant assumes that the authentication method is running and receives feedback based on their actions to confirm this. In actuality, the application reacted to predetermined actions and triggers; no transparent authentication method was actually implemented.

7.2 Study Design

The Transparent Authentication Perceptions (TAP) study is a lab-based, between-groups study [152, p. 74] in which 30 participants were asked to complete seven tasks using an Apple iPhone provided by the experimenter for the duration of the study. The seven tasks were divided into three security levels: Low, Medium, and High, that represented the level of device confidence the device must have before the task is allowed.

7.2.1 Participants

The 30 participants ranged in age from 20 to 58 years old (median = 26.5, mean = 29.4). All participants were mobile device owners currently living in the United Kingdom, and thus had experience with the UK mobile phone network. 60% of the respondents were Android users with various handset models, 13% were iPhone users, 10% used a Blackberry and the remaining 17% used a feature phone (i.e., non-smart phone). 17% of the participants were female and 83% were male. Participants were recruited using convenience sampling methods. Specifically, they were recruited through a combination of email invitations, requests for participation to university classes, and word-of-mouth from other participants to friends and family.

Each study participant was randomly and evenly assigned to either the All, Some or None category. The random nature of this selection is essential in order to avoid study bias and to distribute other possibly confounding influences on the study outcome across all three participant groups.

7.2.2 Apparatus and Materials

Each participant used an Apple iPhone 4 with iOS version 5.1.1 that was provided for their use during the study. It was pre-loaded with the study application and preset with the participant's randomly chosen category and a starting device confidence of Low for all participants. Since the device was provided by the experimenter, it was possible to control the operating system version, as well as other applications and data on the device. This limited the effects of other authentication methods, applications and data that may have interfered if the participant's own device had been used, as was done in the keystroke dynamics and speaker verification studies. The experimenter recorded the interviews, with participant permission, using the Voice Memo application on another iPhone.

After the participants had interacted with the transparent authentication application, they were asked a series of questions in a semi-structured interview in order to collect their opinions on the security levels, perception of barriers and needs for transparent authentication. The interview responses provided answers to the questions that drove this research.

7.2.3 Procedure

The study began with a short questionnaire designed to elicit the participant's age range, gender and whether they currently own and use a smartphone. The participant was then given a short introduction to transparent authentication, and introduced to the Apple iPhone and the custom application they would use for the experiment. The participant was told that a

transparent authentication method was running on the device. This description included a discussion of keystroke dynamics and speaker verification as behavioral biometrics, and the role of explicit authentication (i.e., challenge questions) in transparent authentication. The participant was instructed on how to turn off the authentication and answer the challenge question to override the transparent authentication should they wish to at any point during the experiment. These steps were taken to build a mental model of the intended transparent authentication method, although the actual working of the application depended on the category to which the participant had been assigned.

Since this study was designed in a Wizard of Oz style, there was no authentication system of any type running on the device; the required device functionality was allowed or disallowed based on the settings entered by the experimenter using the interface shown in Figure 7.1. Each participant began the study at the "Low" security level and the category to which they were randomly allocated. Participants were then given an information sheet that outlined the steps for each task they were to perform, as well as instructions on how to answer their challenge question and how to turn off the transparent authentication method. A custom iPhone application was designed for this experiment that was unlike Apple's usual icon approach to tasks and applications; this was done intentionally so that the participant had a sense of using something different than the usual Apple interface.

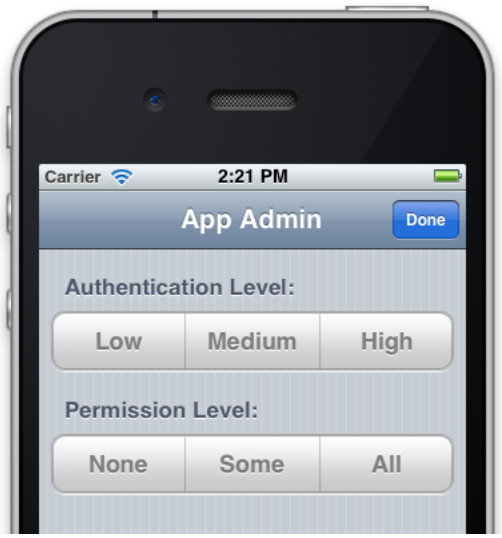


Figure 7.1: TAP Study setup screen. This was not seen by the study participants.

Upon launching the study application, the participant was prompted via an alert box to set the answer to their challenge question (see Figure 7.2a). The challenge question was provided as a backup to the transparent authentication method. In the case where the current device confidence is too low to allow access to a task, the challenge question would be used to authenticate the user and raise the device confidence to the next level.

Next, the participant saw the main "Tasks" screen as shown in Figure 7.2b. It is from this

screen that the participant was able to attempt each task. The participants were given a detailed instruction sheet that they used to complete each task so that each participant completed the tasks in the same order and using the same methods. This was important so that the user perceptions of security and task difficulty were not affected by the order of events. The order of the tasks were from low to high security, and affected whether or not the explicit authentication method was required.

(a) Challenge question alert

(b) TAP application Tasks screen

Figure 7.2: Screenshots of the starting screens for the TAP application.

7.2.4 Tasks

Each participant completed a series of tasks that were grouped into one of three security levels: Low, Medium and High. The security levels directly map to the device's confidence that the participant is the authorized device owner. Since the experiment does not last long enough for the participant to build up a device confidence of sufficient level, the device confidence was initially set to "Low" by the experimenter and the participant was instructed to assume that it was based on previous biometric authentication. The assumption given to the participant was that this device was their own phone and that their keystrokes and voice data had been sampled previously and placed on the phone in order to achieve a device confidence of "Low", and that their ability to complete tasks was dependent on this confidence.

The tasks were chosen to represent common functionality that a device owner may access regularly, as well as for their familiarity to participants. Since one of the study goals was

to determine how easy or difficult the task was to the participants, selecting familiar tasks may mean that any increase in the task difficulty may be attributed to additional steps due to security provision. The tasks were placed into one of the three levels based on the general level of privacy or sensitivity a particular task warranted. For instance, viewing a photo has the potential to disclose more private information than taking a photo, since the device owner has the option of deleting any photos taken by another, but cannot stem the information leak if an unauthorized person viewed or forwarded a sensitive photo. In a production system, the device owner would ideally be able to choose the task authentication level for each function of their device.

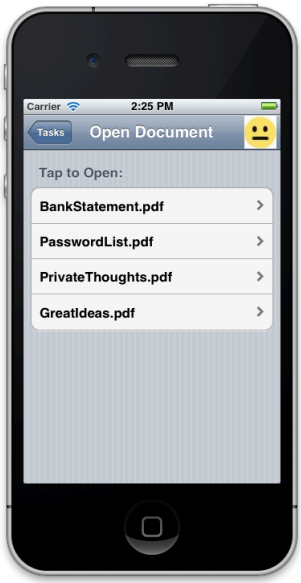
The participants were reminded that the purpose of the study was to assess their impressions of the *security features* of the transparent authentication method as described to them, and not their ability to achieve the tasks, nor the user interface of the application itself.

The tasks were available through the interfaces pictured in Figure 7.3. This interface served as a method of implementing the functionality required by the study, as well as a method of separating the task from the interface. For instance, if the Send Email task is not allowed due to a low device confidence, the decision is made and the participant notified *before* the send email screen is made visible. If the task is allowed, the decision is made in the custom interface and control reverts to the standard iPhone Mail application. Since the study's purpose is not to test the usability or perceptions of the interface itself, the use of a custom, non-standard Apple interface was considered justified.

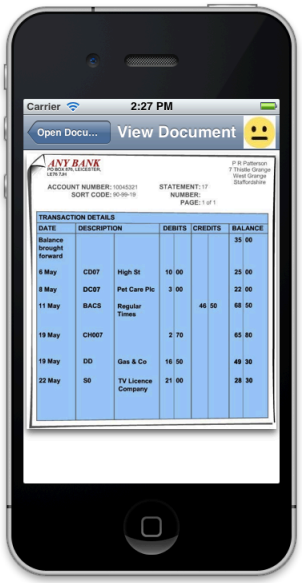
Low Security Tasks

Read Document: Several ostensibly private documents were pre-loaded onto the device. The participant was asked to choose one from a list (see Figure 7.3a), open it and read the contents. The document topics were chosen to create a sense of privacy; while the documents did not actually belong to the participant, they were asked to assume that they did. The documents included a bank statement, private thoughts, and a password file. This task was included in the low rather than medium or high security tasks deliberately in order to determine whether the participants would change the security level due to the perceived document sensitivity. This task was intended to determine whether assigning security levels by *task* was a realistic way of mapping device confidence to device functionality, or if the participants had other preferences, perhaps based on the contents of the document.

Take Photo: The participant was asked to take a photo of a diagram on a whiteboard in the study locale using the mobile device. Taking a photo on a device is not considered a high-security task since it is unlikely to cause embarrassment to the device owner, who



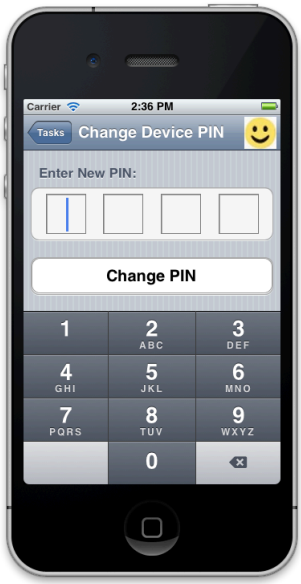
(a) Open Document task screen.



(b) An open document.

(c) Take/View Photo task screen.

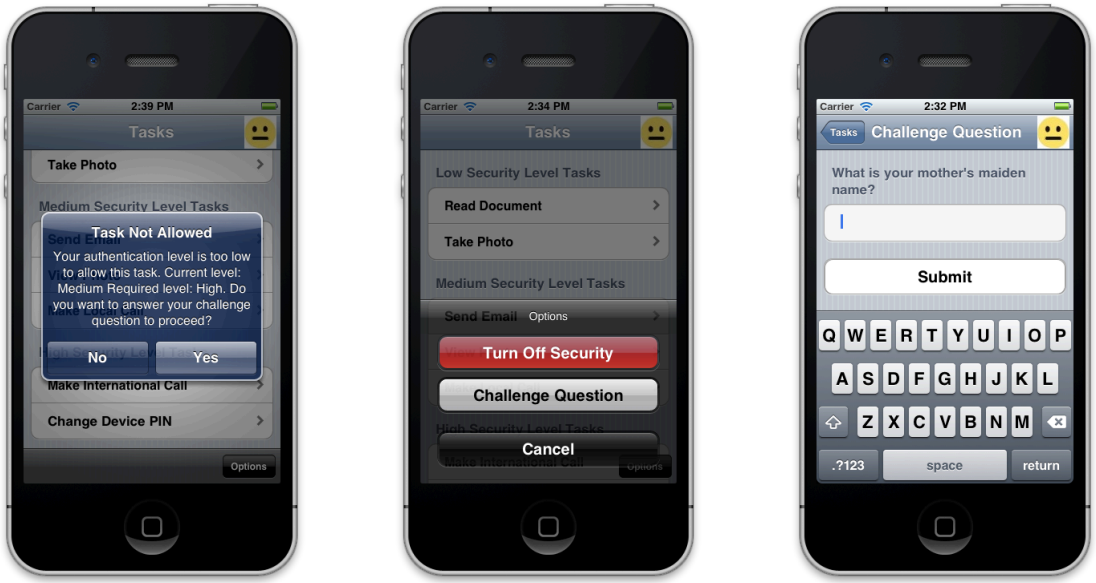
(d) Make local/international call task screen.



(e) Change Device PIN task screen.

Figure 7.3: Screenshots of the individual task screens for the TAP application. The screen for the Send Email task is not shown because it is the standard interface for Apple iPhone.

can simply delete such photos. Exceptions exist, especially in cases of applications where a photo can be immediately uploaded to social networking sites, for example. However, it is envisaged that access to such applications would fall under a higher security level, where access to use of the device camera itself is not a security concern. This assumption is supported by the security on an Apple iPhone under iOS 5.1, since



(a) Task Not Allowed alert.

(b) Options button screen.

(c) Challenge question screen.

Figure 7.4: Screenshots of the support screens for the TAP application.

the ability to take a photo with the device camera does not require entering the device PIN (if enabled). Only the device user has access to the photos, not other photos stored on the device, unless the PIN is entered. See Figure 7.3c for the interface for this task.

Medium Security Tasks

Send Email: The participant was asked to send an email to a particular email address, with text provided by the experimenter. The text was intended to be somewhat private to give the participant a feeling that they would want to prevent others from seeing it. Sending an email was a medium security task because if the device was used by someone other than its owner, sending an email could spoof the real owner's identity and cause embarrassment or other negative effects. This task was also included to provide a way for the user to type during the study in order to provide a biometric match or non-match based on their typing pattern. No biometric classification was performed; either match or non-match was randomly selected after typing. After the task was completed, the participant was told whether their keystroke dynamics biometric was a match or non-match and the device confidence level was adjusted up or down accordingly. The send email interface is not shown here because the standard Apple Mail application was used to support this task.

View Photo: The participant was asked to view a photo, generally the one of the diagram

that had been taken in the "Take Photo" task. This task was intended to get the participant thinking about viewing photos versus taking photos and the security ramifications of others viewing their (potentially private) photos. For these reasons, the View a Photo task was set at medium security. The task interface on the iPhone can be seen in Figure 7.3c.

Make Local Call: The participant was asked to use a custom interface as seen in Figure 7.3d to dial a local phone number provided by the experimenter. The phone number went to a generic voicemail prompt, thereby giving the participant no information as to who they might be calling. They were then asked to leave a message of a private nature. Making a local telephone call is not usually a security concern because they generally have no or minimal cost associated with them, and the caller cannot usually spoof the identity of the device owner simply by making a call, assuming the device owner's voice is known to the call recipient. Some other form of subterfuge (and associated skill level) is required, such as imitating the device owner's voice. However, access to private information may happen if the call recipient is not familiar with the caller's voice. This task was included in the study so that the participant was required to speak and thus (theoretically) provide a biometric sample. As with the "Send an Email" task, the participant was informed whether their speaker verification biometric was a match or a non-match, with the accompanying adjustment of the device confidence level.

High Security Tasks

Make International Call: The participant was asked to use a custom interface as seen in Figure 7.3d to dial a long-distance telephone number that was provided by the experimenter. Dialing a long-distance call may have a high cost associated with it compared to making a local call, and therefore is considered a high security task for the purposes of this study. This task was included to provide another reason for the participant to speak, and therefore have another speaker verification match or non-match, as with the "Send an Email" and "Make Local Call" tasks.

Change Device PIN: The participant was asked to change the device PIN using a custom interface as seen in Figure 7.3e. This task is considered a high security task since changing the PIN in a case where the old PIN is known to others provides additional security, and if someone other than the device owner were to have access to this functionality, they could change the PIN to something not known to the device owner, which may temporarily lock the owner out of the device. This task was included to assess how participants perceive the value of the PIN mechanism and the security it provides.

The participant's device confidence was affected due to two possible scenarios. First, they performed one of the tasks from which a biometric could have been gathered (i.e., Send Email, Make Local Call or Make International Call). As mentioned previously, a randomly chosen decision (match or non-match) either raised or lowered the current device confidence level. For instance, after typing the email the decision was made in the background and the participant notified via an alert box that they had to clear before continuing with the next task. The alert box contained the decision and the effect it would have on device confidence. If the decision was "match", then the device confidence was raised one level (from Low to Medium or Medium to High). If the decision was "non-match", the device confidence was lowered one level (from High to Medium or Medium to Low). The participant was able to see the current device confidence via a series of smiley faces displayed in the upper right corner of the application screens. Figure 7.5 shows the faces and their meanings. This mapping of biometrics to device confidence adjustment was intended to represent the transparent authentication method. In a production system, the alert box with the decision would not be shown, and the device confidence level representation would be more granular. For example, device confidence could be represented as a horizontal bar that has more or fewer ticks in it, depending on the device confidence.

The second scenario in which device confidence may change is when the participant used the explicit authentication feature. The explicit authentication method for this study was a challenge question. If the participant entered the correct challenge question response, the device confidence was increased by one level, much like with a biometric match. If they entered an incorrect response, the device confidence was lowered by one level, as with the biometric non-match. The participant was told whether or not their response was correct via a notification, although this would not happen in a production system. This scenario represents the case where transparency must be foregone in order to increase device confidence to allow a task.

(a) Low (b) Medium (c) High

Figure 7.5: Visualizations of Low, Medium and High device confidence.

Each participant was allocated randomly to one of three categories, which affected their ability to complete the tasks. The participant was able to perform the tasks firstly based on the current device confidence and secondly on their pre-set category, as described below. For example, a participant could be in the Some category and have a device confidence of

Medium, and would be able to perform tasks assigned the Medium or High security levels. The categories were as follows:

None: The participants in this category were unable to complete any task, regardless of their current device confidence. This group is intended to assess the level of frustration seen in a seemingly broken authentication method – one that does not allow task completion. This group is also intended to determine whether the participant chooses to turn off such a method if it gets overly frustrating, and whether they choose to override the transparent authentication with explicit methods when given the chance. This level of authentication can be seen to mimic the first stages of using a transparent method, when the device owner has not yet provided sufficient biometric samples to create a baseline for future comparisons. At this point, the device owner would be blocked from performing most tasks on the device and would have to resort to using explicit authentication to perform all tasks.

All: The participants in this category were able to complete *all* on-device tasks regardless of device confidence. The purpose of this category was to see whether the participant becomes distrustful of the security provided by the transparent authentication method since they are neither challenged nor denied access to data or device functionality. This level is meant to mimic the situation in which the mobile device user suspects the security method is allowing full access to all users; for instance, a user may become suspicious if their fingerprint reader does not occasionally say that their fingerprint did not match the one on file. This group tests all goals as stated in Section 7.1.

Some: The participants in this category were able to complete the tasks that were at their current device confidence and those at any lower confidence, but were unable to complete tasks at a higher device confidence. As such, the application compared the current device confidence to that of their current task, and allowed access if the task level (Low, Medium, or High) was lower than or equal to the current device confidence. The participant could raise the device confidence by answering their challenge question or by having a matching keystroke or speaker biometric result. The purpose of this category is to test questions 2 and 3 in Section 7.1 and to see whether the participant would choose to use explicit authentication in order to complete their tasks. This setting mimics the real design and use of a transparent authentication method, where the current device confidence is matched to a pre-chosen authentication level required for a given task.

The participant's category and their current device confidence determined what tasks were allowed at any given time. Table 7.1 shows the task availability for each category over all

possible device confidence levels. The checkmarks represent tasks that are allowed and the crosses represent tasks that are denied, assuming transparent authentication is enabled.

		Category								
		None			Some			All		
Device Confidence		L	M	H	L	M	H	L	M	H
Task Level	Task									
Low	Read Document	✗	✗	✗	✓	✓	✓	✓	✓	✓
	Take Photo	✗	✗	✗	✓	✓	✓	✓	✓	✓
Medium	Send Email	✗	✗	✗	✗	✓	✓	✓	✓	✓
	View Photo	✗	✗	✗	✗	✓	✓	✓	✓	✓
	Make Local Call	✗	✗	✗	✗	✓	✓	✓	✓	✓
High	Make International Call	✗	✗	✗	✗	✗	✓	✓	✓	✓
	Change PIN	✗	✗	✗	✗	✗	✓	✓	✓	✓

Table 7.1: Task availability by current device confidence and study category. ✓implies participant had access to the task; ✗implies access was denied, assuming transparent authentication is enabled.

While the participants were completing the tasks, the experimenter recorded observations. In particular, the number of times the challenge question was used per task was recorded. Also, the number of times the transparent authentication method was turned off was recorded per task. These values are expected to vary depending on which category the participant was in. Those in the All category, for instance, should not have needed to turn off security or answer the challenge question. The participants in the None group, however, may have tried the challenge question several times before turning off security, and may have tried turning security back on before other tasks. Once the participants had completed all tasks, they were asked a series of questions about their experience in a semi-structured interview. The participants were debriefed after the interview; at this point they were told that no transparent authentication mechanism had been running on the device, and about the three participant categories, including to which category they had been allocated.

The independent variable for this study is the level of transparent authentication the user sees: a high level (the None category), a moderate level (the Some category) or a low level (the All category). The dependent variables were their subjective perceptions of transparent authentication and their subjective beliefs about the security level provided by a transparent authentication method. These were measured using ordinal-answer questions (i.e., Likert scale questions) as well as a semi-structured interview whose results were analyzed using the Grounded Theory approach [172, p. 101] to elicit themes in the answers. These measurements allow linkages between perceptions and stated opinions based on the interview question. These linkages offer answers to the questions posed in the previous section.

7.3 Results and Analysis

The participants attempted all tasks on the device via the custom designed application and in the same order. No participants withdrew from the study, and each participant was paid £6 for their time. The authentication-specific behavior observed as the participants completed the tasks is discussed in terms of the themes that emerged from the qualitative analysis.

Figure 7.6 shows the usual method of traditional security used by the participants on their own mobile device. The Sketch category refers to the sketched password used on Android devices. 27% of participants used a 4-digit PIN, with the same percentage using a sketched password. 30% used no security method, and the remaining 16% used another method. The other methods included encryption, passwords and specialized software that could be used to wipe the device memory remotely in the event it is lost or stolen.

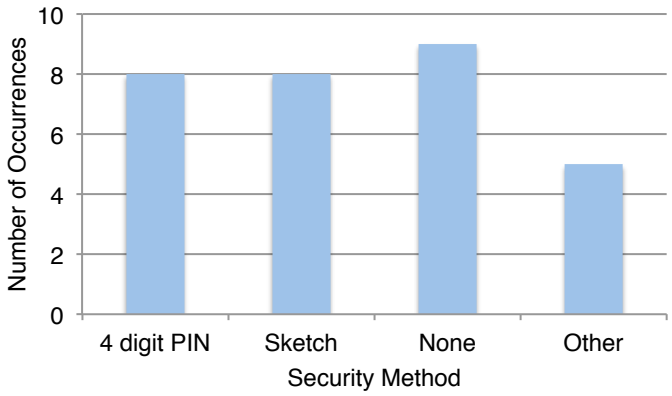


Figure 7.6: Traditional security methods used by the 30 participants on their own mobile device.

The responses to the questions in the interview were used to determine the acceptability of, and perceptions towards, transparent authentication on a mobile device. Statistical significance of the ordinal data questions was determined initially using the Kruskal-Wallis test. This test was chosen for its applicability to non-parametric data with three or more independent participant categories and determines whether there is a statistically significant difference between the three categories used in the study. This test does not indicate between which categories the significance lies. In cases where the Kruskal-Wallis test indicated statistical significance, the inter-category significance was tested in a pairwise manner using the Mann-Whitney test, which is suitable for use on non-parametric data in cases where there are two independent groups, as was the case here.

Some interview questions, as seen in Appendix A, asked a participant to rank a task or perceived level of security. Statistical significance tests were not performed for these results because these tests make assumptions regarding the data that are not supported. For example,

these tests assume that a response of High by one respondent would cancel out a response of Low from another respondent, which is not the case.

The participants were asked why they chose a particular response to a given question. This qualitative data was analyzed using the Grounded Theory approach, which does not depend on pre-selected themes [172, p. 101]. Instead, this approach allows themes to emerge naturally, which is preferable for this work since the results of this study were used to inform the creation of the Transparent Authentication Framework.

Several themes emerged from the qualitative data analysis, as presented in the following sections. The first theme, basis for security level choice, provides an answer to the first stated question of this study (see Section 7.1). It provides an insight into user perceptions when choosing security levels. The second theme, security as a barrier, answers the fourth question about the helpful nature of removing security barriers, as well as the fifth and sixth questions, which ask whether they would use or override transparent authentication and why. The second question, regarding perceived security, is answered by the final theme, which is user perceptions of transparent authentication.

7.3.1 Theme 1: Basis for Security Level Choice

The traditional security methods used by the participants are shown in Figure 7.6. The majority of device owners chose to use a security method on their device, although there were still a number who used no access control at all. Follow-on questions in the interview showed that there was concern regarding the data and functionality on current mobile devices, and that the participants in this study attempted to protect it. One reason given in the interviews for not using access control was the inconvenience of having to enter a password or PIN frequently. Mobile device use is characterized by a bursty use pattern where owners use their device frequently but for short periods of time. Requiring a device PIN prior to each interaction may increase frustration and inconvenience. Indeed, 39% of overall participants gave this as a reason for choosing to forego using access control. Other reasons included fear of forgetting the PIN or password, and perceived susceptibility to observation attacks, particularly for sketched passwords. Forgetting passwords and PINs may be seen as an inconvenience, which falls into the same reasoning behind the choice to not use access control at all. Perceptions of susceptibility to observation may an understanding of the limitations of the security provided by the access control mechanism.

Provision of point-of-entry security such as the methods discussed above protects all functionality and data on the device equally. Transparent authentication allows for the possibility that security levels can be assigned on a per-document or per-task basis; the latter was the assumption made for this study. Figure 7.7 shows the participant responses for the required

security for each task, grouped into High, Medium and Low as an aggregate of the three participant categories.

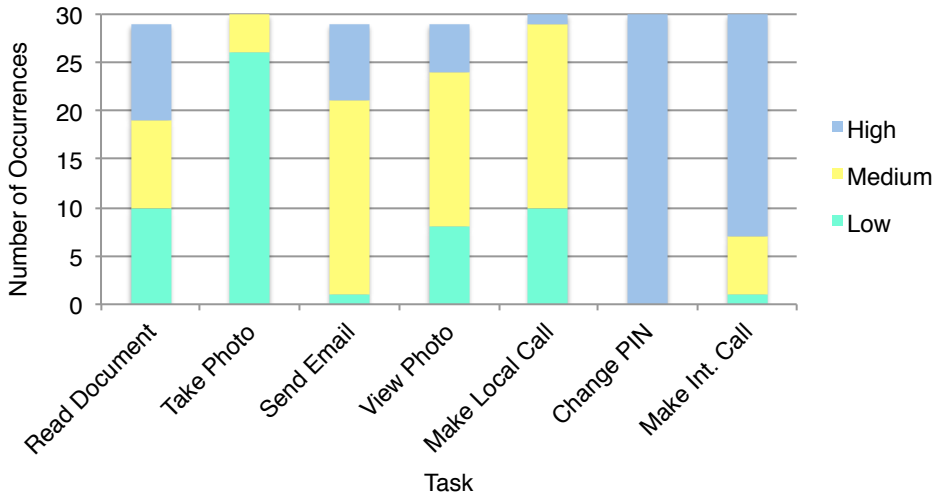


Figure 7.7: Participant choices for task security level

All participants, regardless of category, considered Change Device PIN a high security task. This result indicates that changing PINs was considered a "meta-security" task, in that use of a PIN controls access to all data, most functionality and settings on the device as well as providing point–of–entry security. Some participants noted that control over the device and its functionality belongs to the person who knows the PIN. For example, if the PIN was changed by another person, the device owner would no longer be able to use the device. One participant referred to a PIN-locked device as a "brick" if the owner does not know the new PIN. This comment underscores the uselessness of the device if the new PIN is not known.

Participants did not consider the Take a Photo task to be high security. Taking a photo adds data to the device rather than editing or exposing existing data, and is easily deleted by the device owner. Therefore, this task is not a source of data leakage or privacy concerns.

The Read Document task had a relatively even split between high, medium, and low security. This shows the link between the *contents* or *subject* of the document and the preferred level of security. Participants preferred to have the ability to assign a security level based on the sensitivity of the document's contents, rather than to the task itself. When asked to select a single level when undecided, many participants chose a higher security level with the intention of better protecting the more private or sensitive information. There was a distinction between personal and business-related documents; the former were referred to with the terms "personal", "private", which denote a sense of ownership. Work-related documents, on the other hand, were referred to as "sensitive" and "dangerous", which imply risks associated with their exposure, but not a sense of ownership.

The differences between the preferred security levels per task show that there are a number of

considerations taken into account by participants when intuitively determining the sensitivity of a given task or data. The considerations were major themes discovered through qualitative data analysis of the responses given when the participants were asked why they chose a particular security level for the task in question.

Perceived Risks

The study participants cited the following risks that affected the levels to which they allocated the tasks:

Data Loss or Exposure: is strongly linked to data ownership. For example, participants made a distinction between loss of personal data versus work–related data. Loss of personal data implies loss of reputation or "face" that may be difficult to overcome in the device owner's social circles, but loss of business data could result in loss of a job and professional reputation.

Impersonation: Particularly with respect to sending email, the risk of impersonation was a strong theme throughout the interviews. The severity ranged from pranks by friends who may send a false email to a mutual friend, through examples that included sending negative or derogatory email to the owner's boss, or using the owner's email as a way of "doing evil things" or committing fraud.

Financial Loss: was prevalent when discussing making telephone calls, both international and local. The perceived risk of financial loss was directly proportional to the chosen security level. For instance, international calls were considered more expensive than local calls, and thus were placed in a higher security level. Thus, associating financial loss with a particular task makes it more likely that device owners would take more extreme measures to protect the data or the task.

Loss of Reputation: Usually considered as a secondary risk to impersonation, it was divided into personal reputation amongst friends, and professional reputation. The former held more risk of embarrassment and was a particular concern to younger participants. The risk was humiliation and teasing. With professional reputation, the risks were much greater, including job loss and the inability to gain another job in the same field.

Embarrassment (Misinterpretation of Actions): Strongly related to impersonation and loss of reputation, embarrassment was a risk factor that was associated with many of the tasks. Participants were particularly concerned with embarrassing or compromising

photos and other images, as opposed to emails, text messages, or documents. The embarrassment risk was not in the subject of the photo itself, but with the risk that others may see it, or perhaps pass it onto other via email or MMS.

Identity Theft and Fraud: Identity theft differs from impersonation in that the latter is single instance and ID theft is multiple instances and has much more serious consequences due to the importance of identity in transactions such as banking.

Damage control after data compromise: Once a person's identity is stolen, it can take a significant amount of time to reclaim the identity and to rebuild reputation and credibility as well as things such as credit ratings and credit card ownership. In less far reaching situations, there is an aspect of damage control linked to the embarrassment and reputation risks, since time and effort must go into rebuilding status in both social and professional spheres.

Access to some data or tasks may imply access to others: Coupling of tasks and data access is common on mobile devices. For instance, access to email may imply access to the device owner's address book. It was unclear to many study participants whether protecting one task implied protection of all coupled tasks or data.

Data/Task Sensitivity

If a task or data is considered sensitive, personal or private, the participants in all three categories felt that the device confidence level required to access the task or data should be higher. This also includes the perceptions users have of their own data on the device in terms of the amount and its sensitivity. Many of the participants did not consider their data on-device as important or sensitive, and many believed they had little data on their device. Many participants seemed unaware of the amount and type of data stored on their device, whether placed there by themselves or on their behalf. This finding shows that owners do not understand what information is on the device and may not be able to adequately assess the risks of its loss. For instance, most mobile devices store such personally identifying information as GPS coordinates, phone call timings and recipients, email messages, and text messages, even when it is believed that these have been deleted.

Control over Data or Device

While device owners often misjudged what data was on their device, they expressed a strong preference to control both the physical device itself and the data it contained. Such a finding indicates that device owners have a sense of identity attached to the device and highlights the belief that mobile devices are single-user. This sense of identity meant that the participants

wanted to keep their personal and personally identifying data on the device and within their control. One participant suggested that since the biometric data is already on the device, it is a positive benefit to the device owner to have this data used for security provision:

> "In the past people might have raised concerns about storing that kind of information [keystrokes and voice] on a mobile device, but ...if it's already on there, why not use it to provide additional security? It's practically already recording your voice, and it's already recording what you're typing and things like that, so, I'm not sure the objection of storing that information on a mobile device is valid."

Device sharing, as defined by a device owner allowing another to use their device temporarily, was cited as another reason to assign security levels according to perceived data sensitivity. Participants stated that having public and private folders or memory locations would allow them to share their device without risking sensitive data exposure, although supervision during device use was non-negotiable. This finding shows that electronic security methods may only engender a certain amount of trust, and that techniques such as supervision and physical possession of the device eased security concerns. This latter method was voiced by a participant, as follows:

> "...it never really leaves my pocket, so I don't actually have any real security because I'm scared I'm going to forget the PIN."

The sense of control over the device and data extended to the security mechanism. When asked whether they would consider using a transparent authentication method on their own mobile device, 83% of participants stated that they would, at least on a trial basis. The participants stated that they would "play around with" the method to "see how it works". Such a statement shows the owner's desire to know the security provisions provided, even in a transparent method, and to have control over its use and access to data. Furthermore, it suggests that they may want to understand how intrusive the security provision will be before committing to its use. Reasons for subsequently removing transparent authentication included annoyance, too frequent explicit authentication, or if they believed the method "allowed anybody to access my stuff". Interestingly, many participants stated that their feeling of device and data security was enhanced by barriers in the way of accessing data, although others considered such barriers annoying and frustrating.

7.3.2 Theme 2: Security as a Barrier

Although some participants perceived increased security due to barriers such as explicit authentication, these barriers were more often considered to block their own access to desired

tasks and data. Figure 7.8 shows the frequency of explicit authentication use per task.

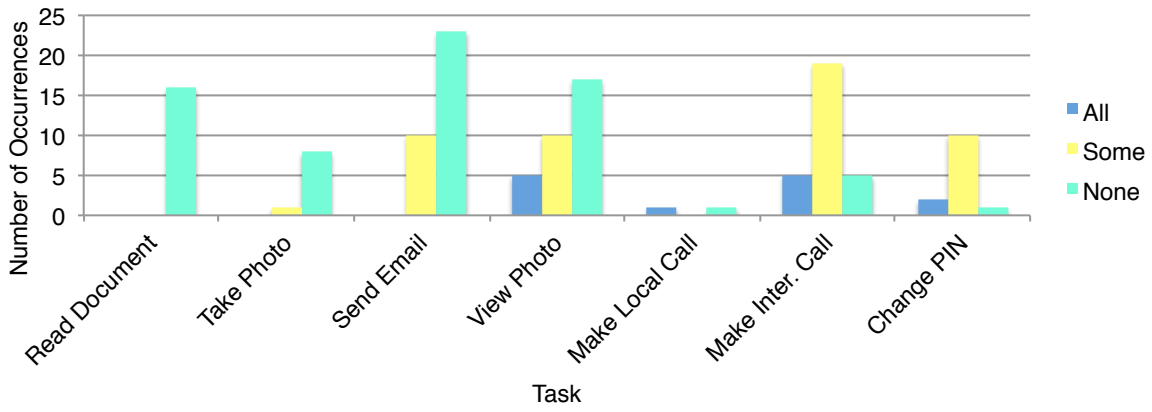


Figure 7.8: Per-task frequency of explicit authentication use for each participant category.

There were significant differences in the frequency of explicit authentication use in all tasks except View Photo and Make Local Call (see Table 7.2). The order of the tasks had an effect on these values, since all participants began the study at a Low device confidence, and thus had access to at least the first two tasks as they were Low security. The exception is the None category, since they were unable to complete any task as long as transparent authentication was on. The significance in explicit authentication frequency per task can be interpreted as the number of barriers presented to participants in various categories; the All category had no barriers at all, the Some category had a moderate number, and the None category had a large number. It is interesting to note that some participants in the All category decided to use the explicit authentication despite having task access without it. This shows that they had a strong mental model of the transparent authentication mechanism, and attempted to work within it.

Task	ρ **Value**	
	Explicit Authentication	Turned Off Security
Read Document	$<$ **0.0001**	$<$ **0.0001**
Take Photo	**0.0436**	**0.0030**
Send Email	**0.0008**	**0.0025**
View Photo	0.1290	**0.0146**
Make Local Call	0.5958	0.1260
Change Device PIN	**0.0009**	0.3679
Make International Call	$<$ **0.0001**	**0.0490**

Table 7.2: ρ values calculated using the Kruskal-Wallis test for frequency of explicit authentication use and disabling of transparent authentication. $\rho < 0.05$ are significant (bolded values).

To determine which categories contained the significant results, pairwise comparisons between the frequency data for the All, Some, and None categories were performed using the Mann-Whitney test, as shown in Table 7.3. The View Photo and Make Local Call tasks have been excluded from Table 7.3 because there was no indication of statistical significance as per the results of the Kruskal-Wallis tests.

Task	**Group**	**Participant Category**		
		All	Some	None
Read Document	All	–	NaN	$<$ **0.0008**
	Some	–	–	$<$ **0.0008**
	None	–	–	–
Take Photo	All	–	0.3681	**0.0347**
	Some	–	–	0.1224
	None	–	–	–
Send Email	All	–	$<$ **0.00002**	**0.0147**
	Some	–	–	0.7066
	None	–	–	–
Change PIN	All	–	$<$ **0.0005**	0.5828
	Some	–	–	$<$ **0.00008**
	None	–	–	–
Make International Call	All	–	**0.0012**	1.000
	Some	–	–	**0.0012**
	None	–	–	–

Table 7.3: Pairwise ρ values calculated using the Mann-Whitney test for number of times explicit authentication was used for the tasks that were significantly different. $\rho < 0.05$ are significant (bolded values). The comparison between the All and Some categories for the Read Document task is NaN because there were no occurrences of explicit authentication for either category.

Most of the categories were significantly different from each of the other categories in terms of the number of times explicit authentication was used per task. The exceptions are when comparing All and None for the Change PIN and Make International Call tasks, and All and Some and Some and None for the Take Photo task. These differences show that barriers presented before allowing tasks were significantly more frequent for Some and All. This represents a potentially annoying amount of intrusion into the participants' completion of tasks, a notion that was supported in the participants' comments. Many stated that they would remove the software if it got "too annoying", or commented that the number of times they had to enter their challenge question response was "frustrating".

The Some category had the largest number of explicit authentication requests, as shown in Figure 7.8, which stands to reason since this category had the least access to tasks out of

all categories when transparent authentication was enabled. The None category had fewer explicit authentication uses because they seemed to learn quickly that using explicit authentication did not help them complete the tasks. These differences and supporting comments show that the device owner's threshold for interruption is relatively low. They also reinforce the idea that users see security as a barrier, and that their usual tasks are their main goal when using a mobile device.

To determine the effect of barriers on security provision, the participants in all categories were able to disable transparent authentication. Table 7.9 shows the frequency with which participants disabled transparent authentication on a per-task basis. The Some category participants did not disable transparent authentication for any task. Their mental model matched the actual working of transparent authentication, therefore they were able to complete all tasks using explicit authentication and biometric matches only. The All category members chose to disable transparent authentication before the tasks that required higher device confidence. The None category disabled transparent authentication frequently for the first task, and increasingly less with subsequent tasks. This shows that in more cases the participants chose to disable transparent authentication and leave it off for subsequent tasks. This finding shows that task completion may have been a more important goal than protecting the device and its data.

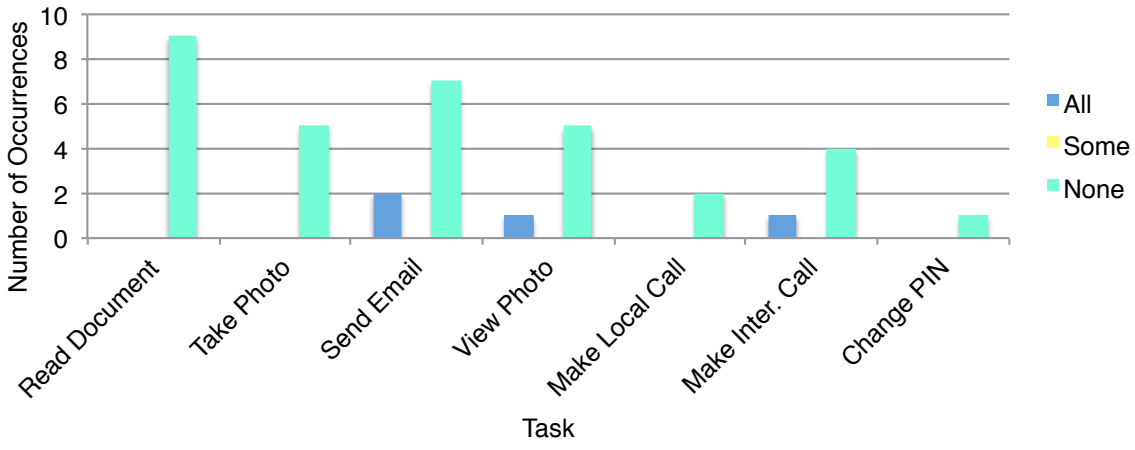


Figure 7.9: Per-task frequency of disabling transparent authentication for each participant category.

Perceiving tasks as the main goal is supported by the significant differences between the None and All and None and Some categories for the tasks in Table 7.4. In both cases, many participants in the All category did not feel the need to disable transparent authentication since all tasks were accessible with it enabled. In the None category, the only way to complete the tasks was to disable transparent authentication, so the difference between these occurrences is understood. Similarly, there would also be many instances in the Some category where disabling transparent authentication aided the participant in completing tasks,

therefore explaining the statistically significant differences between these occurrences and the None category. These results reinforced the finding that disabling transparent authentication, and leaving it off for subsequent tasks, was considered the correct course of action, and that completing the tasks was more important than protecting the information and accessibility of tasks on the device.

An important consideration when assessing the results of disabling security is the fact that the participant was engaged in a study whose perceived goal was to complete a series of tasks on a device that did not belong to them. If the device and the data on it truly belonged to them, the participants may have been more hesitant to disable transparent authentication. It may be that their main goal would not have been to complete the tasks, but to protect their own data. However, these concerns were not addressed during the interview nor during the study design, thus they must be considered but cannot be assessed further.

Task	**Group**	**Participant Category**		
		All	Some	None
Read Document	All	–	NaN	< 0.0001
	Some	–	–	< 0.0001
	None	–	–	–
Take Photo	All	–	NaN	0.0137
	Some	–	–	0.0137
	None	–	–	–
Send Email	All	–	0.1675	0.0318
	Some	–	–	0.0016
	None	–	–	–
View Photo	All	–	0.3681	0.0636
	Some	–	–	0.0137
	None	–	–	–
Make International Call	All	–	0.3681	0.1444
	Some	–	–	0.0336
	None	–	–	–

Table 7.4: Pairwise ρ values calculated using the Mann-Whitney test for frequency that transparent authentication was turned off for the tasks that were significantly different. $\rho < 0.05$ are significant. The two NaN values mark cases where both categories had no instances of turning off security.

The strong theme of security as a barrier is supported by the data. Explicit authentication requests force the user to stop the task they intend to complete and resume it once authentication is complete. The perceived level of frustration with such interruptions was cited as a major reason that participants in this study would consider disabling a transparent authentication method on their device. Thus, transparent authentication methods should aim to

minimize explicit authentication provisions to avoid frustrating or annoying the user, which may result in disabling the security method meant to protect their data and device.

Figure 7.10 shows the participants' perceptions of the task difficulty as an aggregate of all three categories. The findings here further highlight participants' perception of security provisions and their intended tasks as being separate. Most participants considered the tasks either very or somewhat easy. Reasons offered for this determination included that the interface used for the tasks was "simple" or that they were already familiar with the task from previous experience with mobile devices. Tasks that were considered very or somewhat difficult were placed in these categories either due to security barriers ("I had to answer the challenge question twice for that one") or because the task was simply disallowed with transparent authentication on. These findings show a disconnect between security requirements and the task at hand, and that the frustration with security is considered separately from the task at hand. Furthermore, the number of barriers, which varied from category to category, does not have an effect on participant opinions of the ease of tasks, while the responses to other interview questions indicate a higher level of user frustration with transparent authentication from those participants in the None category. This can be summarized with the following quotation from one participant in the None category:

> "...when it works right, then you'll be able to use it without having to unlock it all the time... I do think that, you know, the fact that you have to keep upgrading your level sometimes is a nuisance, but if the system knew more about you, then that wouldn't be a problem."

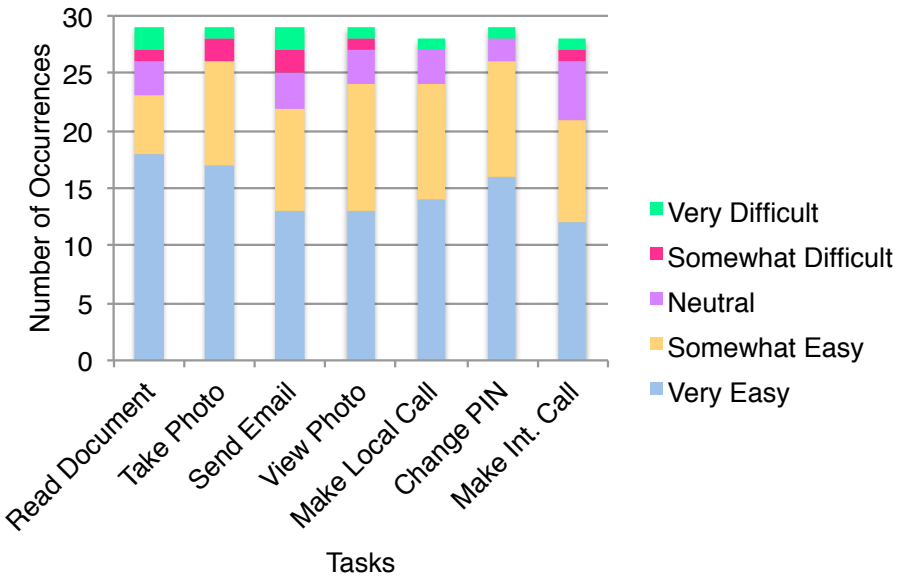


Figure 7.10: Participant perceptions of task difficulty over all categories.

One of the main reasons for the amount of frustration felt when security provision was seen as a barrier was lack of access to the data on the device. A sense of protection for the device and data may be linked to the provision of the very barriers that cause frustration. Figure 7.11 shows the participants' perceived levels of data protection provided by transparent authentication, per category. Participants in the All category thought the data was poorly protected since they indicated an answer higher than neutral in only two cases. This category had the fewest security barriers with which to contend. Conversely, more than half of the None category members, who had the most security barriers, considered the data very or somewhat well protected. The Some category members ranged somewhere between the All and None extremes. They had a moderate number of security barriers, and largely considered the data either somewhat protected or not protected, but never very well protected. These results indicate that barriers, while annoying and frustrating, also provide a sense of security and data protection. This can be related to the device owner's mental model. If the idea of security provision can be communicated in another manner besides difficulty in data access, then the frustration created by these barriers may be reduced while keeping the *perception* of security high.

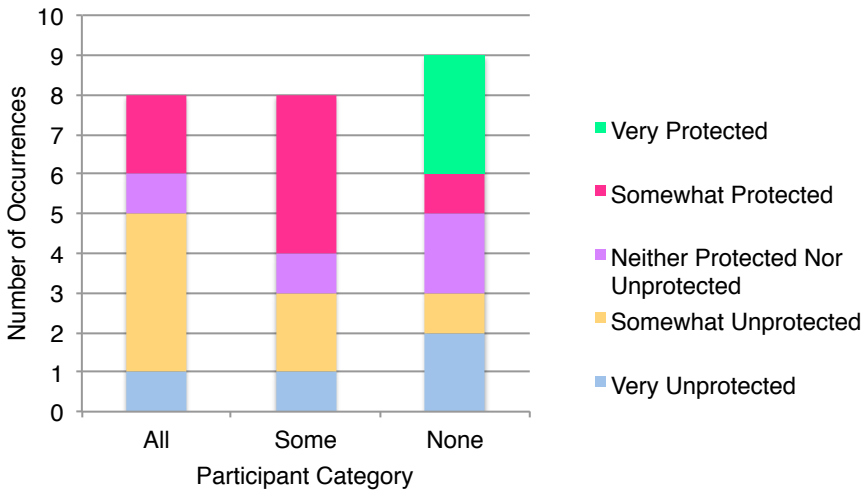
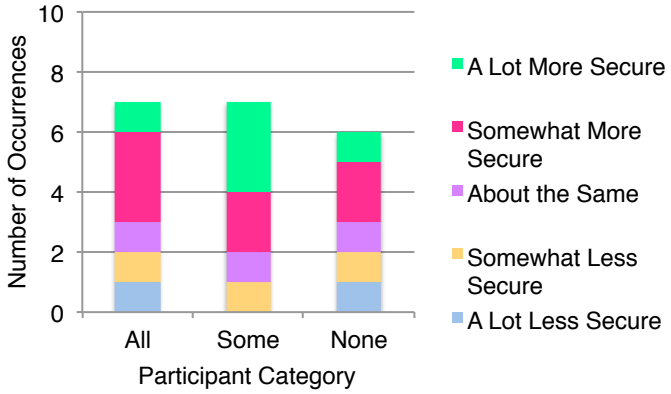


Figure 7.11: Perceptions of data protection provided by transparent authentication. Each category had 10 participants; some declined to provide a specific value, as shown by the shorter bars.

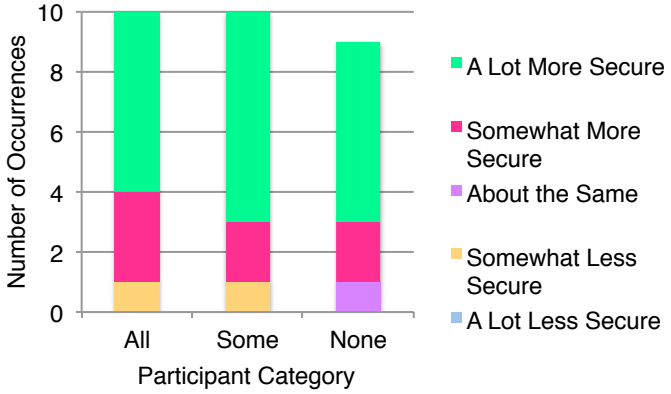
7.3.3 Theme 3: Perceptions of Traditional and Transparent Authentication

Despite the number of participants who chose not to use security provisions on their own device, the participants generally felt that security delivered benefits. Figure 7.12 shows

participant opinions on transparent authentication provision compared to either their current mobile device security method (see Figure 7.12a) or to no security at all (Figure 7.12b).



(a) Transparent authentication compared to owner's current method.



(b) Transparent authentication compared to no security.

Figure 7.12: Comparison of device security. Several participants declined to provide a response, which is why the bars do not each total 10.

These results show that these participants believe that "something is better than nothing" in terms of security provision. This, however, does not explain the actions of those participants who chose to use no security at all on their device. Other things, perhaps the barriers provided by explicit authentication methods, discourage them from using security on their devices when they seem to believe that it is useful. This theme can also be seen in the previously stated opinions on PIN use. Participants saw the PIN as a powerful over-arching security method with a direct link to the provision of applications on their device. However, many of the participants explained their avoidance of PINs in terms of fear of the negative consequences that result from forgetting it. In most cases, participants feel that security provision is improved with transparent authentication as compared to either traditional security methods or none at all. This does not, however, answer the question of whether they

find security provisions necessary, or whether they would use them if provided. Answers to these questions can be partially provided by examining what inclusions participants found important in a transparent authentication method.

7.3.4 Theme 4: Suggestions for Transparent Authentication Functionality

Several areas of improvement were identified by participants. These preferences have been fed into the Framework design to provide a more useful application that meets user needs. The suggestions are as follows:

1. Assign required device confidence on a per-task or per-folder basis, in addition to by task or application. Have pre-set values that can be changed by owner to reduce initial setup effort.
2. Minimize the number of explicit authentication interruptions as much as possible as these are considered frustrating and intrusive.
3. Keep the owner's data on the owner's device. Do not share it with others, or remove it from the device in order to implement a security mechanism.
4. Minimize effort for frequent tasks. This can be managed by allowing the device owner to select a lower device confidence for tasks that are accessed frequently.

7.4 Study Limitations

The limitations to this study are less about the type of device used since this was controlled by the experimenter, and more about the participants and the experimental design. The limitations and their potential effect on the results are listed below.

Subjective Nature: The majority of the data gathered in this study is of a subjective nature; it is the participants' opinions and perceptions and is thus subject to their own beliefs and knowledge. It may be that the same study conducted on a larger or differently populated group may result in some differences in participant opinions and thus affect the study results.

Application Design: The application used in the study does not use the actual Apple interface for the tasks included in the study. This is considered acceptable because the study was not designed to test or gather participant opinions of the Apple interface.

However, it may have added a bias to the participant opinions due to their belief that the application as designed was actually the transparent authentication method.

Device Ownership: The participants in this study did not use their own device, therefore there may be a disconnect between the security level perception and the data on the device, since the data did not belong to the participant. This means that the participant might consider the data adequately protected because its loss would not affect them personally.

Suspension of Disbelief: Although the participants were initially told that a transparent authentication method was currently running on the device, this was not actually the case. Therefore, the interactions with the study application may not accurately mimic a production transparent authentication method. This should be taken into account when assessing the study results because participant opinions may change due to changes in the system itself. One example of how this might affect the results is that the device confidence is intended to increase and decrease in a much more granular fashion. In this study, a biometric result or challenge question answer raised or lowered the device confidence by one level: low to medium, medium to high, and vice versa. The Transparent Authentication Framework described in Chapter 3 states that a single biometric result or explicit authentication attempt would have a small effect on the device confidence; it could well be that such a small change would mean that the device confidence remains at its current level, although at a slightly higher point. For example, if the Low level was defined to be less that 25% device confidence and the device was currently at 24%, a correct biometric match might raise the device confidence to 26% thereby moving overall device confidence to Medium. However, if the device was at 15% device confidence when the correct biometric match was seen and moved to 17%, the overall device confidence would remain at Low, although at a slightly higher level than previously.

Participants: The study participants were self-selecting in nature, since each person was invited to participate by the experimenter. 90% of the participants were either computer science academics or involved in technology-related jobs, such as IT support. This is a potential limitation because the participants no longer represent a subset of the target user group, which is "all mobile device users". The participants instead represent a small subset of the target group, and one that may be more aware of security concerns and mitigation techniques than the average smartphone owner. This could potentially affect the study results since the opinions of a technically-minded and security-aware group may differ from others.

Experimental Error: While significant effort was put into planning the study and develop-

ing the required iPhone app, there were several places where experimental errors were seen that might affect the study outcome. For instance, biometric matches and non-matches were reported after the participant had completed the "Send Email", "Make Local Call" and "Make International Call" tasks. These were intended to simply be a random choice between match and non-match since the participant's biometrics did not exist on the device in order to provide a baseline biometric for actual matching. This was tested during application development and outcomes were observed to be sufficiently random. However, during the study no participants experienced a match result; only non-match results were produced. This is likely due to a seeding issue with the random number generator used to select either match or non-match. Since the application was restarted on the device prior to each participant's use, the seed was likely the same each time and thus only produced non-match results. This may have had an effect on the participants' opinions of the transparent authentication method because they may perceive that it never matches, and thus is not trustworthy.

7.5 Summary

This chapter provided details of the Transparent Authentication Perceptions (TAP) study, a between-users study ($N = 30$) that was designed to elicit user perceptions and opinions of a transparent authentication method like the one described in this work's Transparent Authentication Framework. The study results, as analyzed using both quantitative and qualitative methods, were used to provide impetus for future research and as a method of informing the framework design to more closely match users' wants and needs.

Chapter 8

Security Discussion

This chapter outlines the threats and the vulnerabilities that could be exploited to compromise the Transparent Authentication Framework. Despite the assertion of some researchers that mobile operating systems are sufficiently secure [173], there are still threats that should be considered when using the Framework as a model for transparent authentication. Biometrics, in particular, have their own classes of threats, which are discussed in terms of how they impact the Framework. Mitigation techniques are discussed for each attack, although an attacker may still be able to take advantage of a vulnerability despite mitigation. This list is not (and is not intended to be) exhaustive; new attacks arise daily and an implementation of the Framework will have to adapt to emerging attacks as they are discovered.

8.1 Attacker Capabilities

In order to focus on attacks that are specific to the design of the Framework, the attacker's capabilities are outlined below:

- The attacker is physically near the device and its owner, and can observe their actions. They may be able to take physical possession of the device through theft or temporarily if the device owner leaves it on a table or desk, for example.
- The attacker has knowledge of the Framework and its component processes and data. This assumption is similar to the conventional threat models in cryptography where the algorithm details are not kept secret from attackers.
- The attacker does not know which specific biometrics are used in the particular Framework implementation. This knowledge does not come with knowledge of the Framework itself since it is designed to include several biometrics that are not explicitly specified in the Framework.

- The attacker does not have access to the closed 3G network, nor any secured wireless network that the device owner may connect to with the device. This assumption is justified because it requires a separate set of attacks that are not specific to the Framework.

In addition to the attacker capabilities described above, the following assumptions are made when considering the attacks discussed in the remainder of this chapter:

- Generic attacks that apply to any application on a mobile device, and that are not aimed specifically at the Framework, are out of scope. These attacks can be mitigated in ways beyond the Framework design. These attacks include hardware- and software-centric attacks and device-independent attacks via the wireless networks.
- Attacks that require a malicious version of the application created based on the Framework are not considered. This form of attack also has its own mitigation techniques, and such an attack cannot be mitigated through adjusting the design of the Framework itself.
- The data created and used by the Framework (i.e., biometrics and their decisions) remain on the device. They are not transmitted or stored off-device at any time.

The following sections describe some of the possible threats to the Transparent Authentication Framework. Figure 8.1 shows an overview of some of the relevant Framework functionality, and shows what parts of the Framework could be targeted.

8.2 Social Engineering Attacks

Social engineering attacks include any means of manipulating or deceiving a person into revealing information that might be used by an attacker to gain access to a protected system. While it might seem simple to reject attempts to gather personal information, it is often easier for an attacker to simply ask for the information they require rather than attempting other, more complex attacks [175]. In terms of the Transparent Authentication Framework, social engineering attacks involve gaining a copy of the device owner's biometrics, such as typing and voice patterns, in order to reuse them to spoof the device into allowing access to an impostor. Methods include calling the device owner and recording a conversation, or directing them to a website and gathering typing patterns via a web form (phishing). In both cases, the device owner would not necessarily link the request for information (speaking or typing) to an attack on the device security.

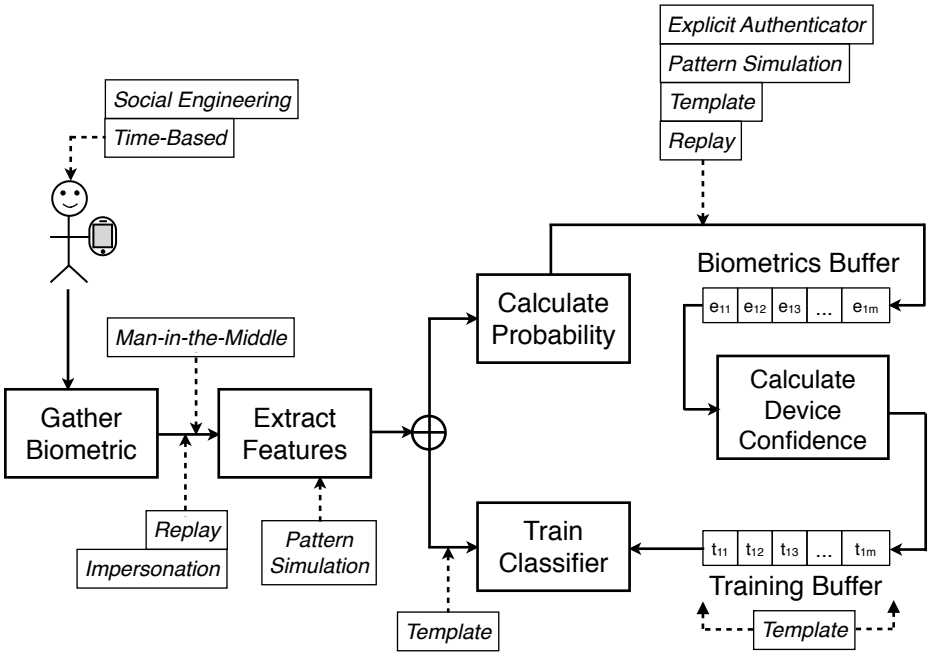


Figure 8.1: The Transparent Authentication Framework and the relevant attacks (shown italicized and in boxes). Adapted from [174].

Social engineering can take another form within the Framework: tricking the device owner into overriding the security by entering the explicit authentication answer. This threat includes methods of fooling the owner into turning off the device security (Framework-based or other) or into continuously providing positive owner samples for classification while the actual device user is an impostor (e.g., speaking near the microphone or revealing the secret knowledge that may be used for explicit authentication). Mitigation techniques include enhanced user education, using multiple secrets or biometric methods for explicit authentication, and encrypting the device data when it is not being used. Use of these techniques involves balancing the mitigation against performance penalties in terms of memory, processor power and battery usage. This type of threat requires the attacker to know what biometrics are being used for authentication, which may not be easily determined simply by observing the device owner.

8.3 Explicit Authenticator Attacks

These attacks, as with explicit authentication use outside of the Framework, take advantage of the weaknesses in the method used for explicit authentication. While the method is not specified within the Framework, it may still represent a security weakness since no security provision is without flaws that may be exploited. Suggestions for the explicit authenticator include challenge questions, a password or PIN, or a biometric that is explicitly gathered,

such as fingerprints. Challenge questions, passwords and PINs are secret-knowledge techniques that are vulnerable to social engineering attacks in which the legitimate owner is tricked into revealing the secret. They are also vulnerable to brute-force attacks in which all possible combinations of secrets (i.e., letters, numbers and special characters) are tried until the correct combination is found. If the explicit authenticator is a biometric, it is vulnerable to the same attacks as discussed in Section 8.5. Mitigation techniques depend on the chosen explicit authenticator, and thus are not described here.

8.4 Time–Based Attacks

These attacks include any that are associated with timings, such as where an impostor uses the device more than the owner and thus trains the classifiers to recognize him rather than the owner, or when the owner has achieved a high device confidence, then the device is lost or stolen. The thief would then be able to perform high level tasks for a period of time before the device realizes that it is not being used by the device owner. This window of opportunity could be sufficiently long that serious loss of data or intrusion results.

A possible mitigation technique for this type of attack is to require use of explicit authentication methods after a set number of contiguous biometric non–matches. The number of non-matches can be selected by the developer or device owner. It is important to set the number low enough that attacker access is limited as quickly as possible, but high enough that the device owner is not locked out unnecessarily, as may happen if a biometric false reject takes place.

8.5 Biometrics–Based Attacks

Biometrics are not secrets, and thus they are vulnerable to being captured, copied and forged. The risk of this is enhanced by the fact that biometric traces are left while doing other tasks. For instance, fingerprints are left on every surface touched and it has been known for quite some time that they can be collected and possibly reused [176, 177]. More germane to modern biometrics systems is that captured latent fingerprints can be used to spoof fingerprint biometric systems [178].

The literature on biometric vulnerabilities has identified some overarching categories, including impersonation, replay, transmission and data simulation attacks [174]. They are applicable to both behavioral and physiological biometrics, but many of the attack vectors are more applicable to the latter. In each case, they are discussed in terms of the Transparent Authentication Framework in particular, with mitigation techniques suggested where

possible.

8.5.1 Impersonation and Replay Attacks

Impersonation attacks, also called *spoofing*, involve stealing the device owner's biometric pattern and replaying it to gain access to the secured system. With physiological biometrics, the norm is to artificially recreate the data and reuse it (i.e., a fake finger or high–resolution eye image). Spoofing behavioral biometrics usually requires mimicry rather than reusing biometric artifacts, although behavioral biometrics are susceptible to both. In all cases, impersonation attacks require the attacker to capture a biometric sample from the legitimate device owner, either for reuse or to use as a template for creating new samples. Impersonation and replay attacks are shown between gathering biometrics and creating the feature vectors in Figure 8.1, and also when adding new biometric event objects to the biometrics buffer.

In the Transparent Authentication Framework, an attacker could launch an impersonation attack by gathering the biometrics used. For example, the attacker could record the device owner's voice while making a call or saving a voice memo. This voice pattern could then be played back at a later date in order to provide a positive voice sample and thereby increase the device confidence. The owner's voice pattern may also be gathered via social engineering efforts in which the device owner receives a call and is engaged in conversation for the purpose of recording their voice. Possible countermeasures include sampling the spoken words, inflections, background noise, and duration of the recording for matches to previously seen samples. This countermeasure is unlikely to be viable, however, because such functionality is quite advanced and unlikely to run efficiently on a mobile device due to resource constraints.

A further example of the type of impersonation attack that may be possible within the Framework is if the attacker attempts to mimic the device owner's typing pattern. Keystroke biometrics are not trivial to mimic. While it is possible that an attacker may watch the user typing and attempt to mimic their pattern, it is unlikely that they will be successful, particularly without a way of practicing that includes feedback on whether they are correctly matching the device owner's pattern. Furthermore, one study has shown that attempting to mimic a legitimate user's typing pattern has little or no effect on the pattern matching results with keystroke dynamics [179]. Spoofing keystroke dynamics may be more likely to succeed when using keylogging rather than mimicry.

Keylogging is an attack where a user's typing is captured and logged, often in an offsite location, although it can also be done on the device itself. Keylogging is used to determine what a person has typed, and is a form of privacy invasion. In terms of the Framework,

if keystroke dynamics are used as a biometric, keylogging potentially could be used in a replay attack where the attacker logs keystrokes and plays them back. This implies that the keylogger must also track the metrics used in keystroke dynamics: inter-key latency and key hold time. The attacker would then have to ensure that the keys were played back in a manner that resembled real typing. They could not, for instance, attempt to enter a long series of characters, complete with key hold time and inter-key latency values, all at once. This would likely be detected by the biometric system. It is more likely that an attacker could use a keylogging program that gathers not only the keys but their metrics as well, and use this information to create new Event Objects that can then be injected into the event object buffers as legitimate owner patterns. This attack would require detailed knowledge of the specific format of the Framework used by the developer, including specifics about the Event Objects and their buffers. A simple mitigation technique includes preventing access to the on-device memory used by the Framework implementation by other applications.

Impersonation and replay attacks can also take advantage of biometric inter–subject similarities. Such an overlap in the feature space between users [57] allows for pattern matches even when the device user is not the owner. One example of this in the proposed biometrics for use with the Transparent Authentication Framework is in typing patterns. There are relatively few ways of typing distinctively, as evidenced by the results of many keystroke dynamics studies discussed in the background provided in Chapter 2. Therefore, an attacker may find a pattern that is similar enough to that of the device owner that the Framework is spoofed into thinking that the device owner is using the device, although the likelihood of this is expected to be low.

An impersonation-style attack that is related to inter–subject similarities could be launched by an attacker who uses the device sporadically. The attacker could type or speak into the device frequently enough that his pattern becomes known to the biometric classifiers, but not so often that he is discovered as an attacker. Over time, the attacker increases the frequency of his device use compared to the device owner, until the classifiers identify the attacker's patterns as those of the legitimate device owner. This attack would be easier to launch and more likely to succeed if there were inter-subject similarities between the attacker's and the device owner's typing patterns. Alternatively, the attacker could imitate the device owner well enough to begin the process, and slowly alter their typing pattern until they are typing as themself. The success of this attack depends on the device owner allowing the attacker to use their device, or the attacker being able to use the device frequently enough to provide sufficient samples in some other way. This attack can be mitigated by not allowing others to use the device.

Replay attacks can be mitigated using liveness testing techniques. Such techniques disallow replay of biometric patterns by ensuring that the supplied pattern is from a live person at the moment it is supplied. For example, fingerprint liveness techniques can check for bloodflow

in veins while scanning the fingerprint. In the Framework, liveness techniques for keystroke and voice patterns could be employed, although it is unlikely to happen within the constraints of battery life, processor speeds and memory availability.

8.5.2 Pattern Simulation

A pattern simulation attack is one in which the attacker generates patterns that are very similar to those of the legitimate owner, and uses these to gain access to the protected resource. This is a form of algorithmic mimicry, or *digital chameleons*, and has been suggested as an attack on behavioral biometric systems [180]. In this case, the mimicry is undertaken by a computer program.

The difference between pattern simulation and impersonation attacks is the source of the pattern. In the latter, the attacker obtains a real pattern from an authorized user and replays it, where in the former the pattern is created in an automated manner, possibly from using legitimate patterns as a template. The difference, although subtle, is important since a simulated pattern would not have the same hash value as a legitimate pattern, and thus a hash-based mitigation technique would fail. Pattern simulation attacks are shown during feature extraction and adding to the biometrics buffer in Figure 8.1. Mitigation techniques for pattern simulation include ensuring that the patterns come from the device itself, and protecting against phishing and eavesdropping, which may be used to gather legitimate patterns.

8.5.3 Man–in–the–Middle Attacks

A man–in–the–middle attack is defined as a form of eavesdropping in which the attacker relays messages between two people, computers, or processes. In this attack, the two original victims believe they are still communicating securely with each other. In terms of the Framework, a man–in–the–middle attack could take place at several points (see Figure 8.1), including between gathering the biometric and feature extraction. Specifically, the biometric can be captured and replaced with another before it is used for feature vector creation.

Man–in–the middle attacks to the Framework would require access to the memory and communication channels between the discrete parts since the biometrics do not leave the device and the connection between gathering and using the biometric is all on–device. Such access is limited on some platforms. For instance, the Apple iOS operating system does not allow applications to overwrite or access memory that is not assigned to that application. In essence, this type of attack would likely have to be done with a malicious version of the software.

Mitigation techniques for man–in–the–middle attacks include disallowing access to memory and communication channels, verifying the identity of the communicating processes, and using secured cryptographic protocols to protect the gathered biometrics and feature vectors. These defences are relatively expensive in terms of memory and processor power and are thus unsuited to a mobile device environment. Furthermore, continuously encrypting and decrypting data throughout the Framework's authentication process adds a significant amount of time to the device confidence calculation, which may become annoying to the device owner. Fortunately, man–in–the–middle attacks are considered unlikely on a closed system such as the Framework.

8.5.4 Template Attacks

A biometric template attack takes place when the information used for comparison to a newly gathered biometric is altered to allow authentication of an impostor. The Framework refers to the biometric template as the trained classifier, or alternatively, the set of event objects in the training buffer. In this type of attack, the training event objects themselves may be replaced, or the trained classifier may be replaced with a pre-trained substitute that has been trained on the attacker's patterns, or the entire training event object buffer may be replaced, as shown in Figure 8.1. In the latter case, periodic classifier retraining would replace the current model that represents the device owner with one that represents another.

To mitigate template attacks, protection must be given to the memory and communication channels between the Framework components. Furthermore, event objects that have not originated from biometrics gathering should be rejected. Since the data does not leave the device and all calculations are expected to take place on–device, it is unlikely that such attacks would be successful.

Mitigation techniques for biometrics–based attacks usually involve either watermarking, encrypting or hashing the biometrics themselves. It is unlikely, given the environmental constraints, that the Framework could support such processor- and memory-intensive operations, particularly on such large amounts of data. However, it is possible that the individual feature vectors used may be protected via these techniques, particularly if a simple encryption technique is used.

8.5.5 Multimodal Biometrics

Using multimodal biometrics is itself a form of mitigation against biometric replay attacks since any attacker would have to provide more than one type of authorized biometric simultaneously in order to create a multimodal biometric [174]. The transparent nature of the

authentication provided by the Framework makes spoofing biometrics more difficult since it is not obvious to observers what biometrics are being gathered. Furthermore, each application that uses the Framework as an authenticator may employ different biometrics, and could replace weak or subverted methods with others as needed. In terms of the Framework, using multimodal biometrics is only a partial mitigation to replay attacks since the Framework also allows use of single biometrics in cases where only one type is available. Use of several biometrics to create multimodal biometrics within the Framework would help use this mitigation technique to its fullest potential.

8.6 Summary

Many types of attacks are specific to mobile device environments. While the Transparent Authentication Framework is vulnerable to these attacks to a greater or lesser extent because it runs on a mobile device, there are also specific attacks that exploit the Framework's structure and components. These have been discussed in this chapter, and mitigation techniques have been suggested for each attack. Many of the suggested mitigation techniques rely on standards such as cryptography, user education, and use of hash functions for software. These may not be worth pursuing in terms of the cost-benefit tradeoff.

In general, the Transparent Authentication Framework is vulnerable to many mobile device-based attacks, and may use the mitigation techniques for these as necessary. The Framework is no more susceptible to attacks than other mobile device software, with the possible exception of biometrics-specific attacks. Many attacks depend on the mobile device environment being used; thus, a security risk assessment should be undertaken before implementing software based on this Framework.

Chapter 9

Conclusions and Future Work

This dissertation has provided details of the Transparent Authentication Framework, including the design, candidate biometrics and a perception study carried out to assess user acceptability of the mechanism. The Transparent Authentication Framework delivers transparent, continuous authentication on mobile devices by relating device confidence to tasks and data on the device. The Framework provides *transparent* authentication by using behavioral biometrics that are gathered in the background. It provides *continuous* authentication by recalculating device confidence whenever biometric samples are available.

To conclude this dissertation, design considerations for the Transparent Authentication Framework are provided. The purpose of providing these design considerations is to inform future iterations of the Framework and to highlight issues in transparent authentication design. The research contributions this dissertation has made are then related to the hypotheses and research questions that define this work. Finally, areas for future work based on the Transparent Authentication Framework are discussed.

9.1 Motivation Revisited

Three core considerations motivated this research. They have been addressed by the Framework in the following ways:

The Password Problem: The Framework may reduce the need for explicit authentication methods such as passwords and PINs by repositioning authentication provision as a background task. This provides the device owner with fewer chances to subvert secret knowledge techniques by using weak or shared secrets. Furthermore, the Framework provides a nuanced approach to security provision that goes beyond point-of-entry security to allow users to map device confidence to tasks and data on the device. This further reduces the reliance on typically weak passwords and PINs.

Disconnect between Mental Models and Password Security: The Framework allows for mapping between device confidence and specific functionality and data on the device. Allowing the user to control the mappings may encourage creation of a stronger mental model, as well as increased knowledge of the data that is stored on the device. Applying a minimum device confidence encourages the owner to assess the risks they associate with the data or functionality, and further supports a strong mental model of the provided security.

Inflexibility in Authentication Policy Creation: With the nuanced approach to security provided by the Framework, the device owner in essence creates their own security policy. Reducing the reliance on often weak secret-knowledge techniques may mean that there is less reason for organizations to dictate policies that encourage circumvention. The Framework allows device confidence to be set on a per-document basis, which allows for increased protection for sensitive documents, rather than relying on the same password to protect everything on the device.

The motivations stated here are further supported by a series of design considerations for the Framework that will help it meet device owner needs.

9.2 Framework Design Considerations

Continuous, transparent authentication is not intended to be the only choice for security provision on mobile device platforms. There will be cases in which other methods, such as graphical passwords, physiological biometrics, and passwords and PINs are more applicable choices. Continuous, transparent authentication has the potential to provide a secure environment that allows security provision to remain largely in the background, and to help alleviate the memory and task load of the device owner. In this way, it becomes another tool in a developer's authentication and security toolbox. Through this research, several design considerations have been identified when considering generic transparent authentication. Through their identification, a broader picture of authentication design considerations can be seen. The design considerations are as follows:

9.2.1 Basis for Security Level Choice

Consideration 1: Transparent authentication methods should allow the user to select appropriate required device confidence levels for both data and tasks or applications.

The choice of these methods and levels are individual, and attempting to apply firm levels may make the Framework useful in fewer cases. Default settings should be provided for particular documents, images, and folders; the device owner should have the ability to adjust the levels to suit their own data and comfort levels. Allowing such choice provides a basis for a mental model of the security provided, and gives not only control over their own data and device, but also a feeling of understanding the Framework so device owners can work within it rather than against it.

The Framework allows for this by setting default device confidence levels for all tasks and data, and allowing users to change these levels at their convenience. The default levels can be set initially by the application developer who uses the Framework, taking into account the specific needs and risk levels of the application under design.

9.2.2 Security as a Barrier

> **Consideration 2:** Transparent authentication methods should work in the background, relieving the user of much of the repetitive provision of knowledge-based authentication methods, while also provide a sense of security and robustness.

One interpretation of this consideration is the balance between providing too many barriers, which are seen during explicit authentication methods such as passwords, PINs and challenge questions, and providing too few. The former may be annoying, frustrating, and provide a reason to stop using security methods. The latter, on the other hand, may reduce the device owner's sense of the security provided by transparent authentication. Barriers could be provided through periodic explicit authentication and reports of biometric non-matches. One result from this research is that they should not be too frequent; a difficult goal, to be sure, since the ideal frequency must vary from person to person.

The Transparent Authentication Framework has been designed to provide continuous, transparent authentication. By definition, it works in the background, although early interaction with the Framework is characterized by a higher number of explicit authentication requests since the Framework is still learning the device owner's biometric patterns. The biometrics, while left to the application designer, should take advantage of backgrounding to gather information unobtrusively. The two biometrics suggested in this research, keystroke dynamics and speaker verification, support background data gathering since they take advantage of regular tasks performed on mobile devices. The sense of security associated with the Framework is an area of future work, although this research has indicated that device owners depend on this idea of robustness in order to continue using security provisions beyond initial testing.

Consideration 3: Transparent Authentication methods should take into account the user's mental model of the security provided, and work towards enhancing the model to bridge the gap between explicit and transparent authentication.

Users create mental models, an internal understanding of how something works, of systems with which they interact [181]. These mental models help the user understand how to interact with a given system, and allow them to begin to build trust in how the system will react. In security, mental models help with risk perception and communication [182], which is informing the user of particular security risks that exist with the use of an application[1].

This research found that the mental model of the Framework determined whether study participants chose to complete their tasks within the Framework or turn it off. Furthermore, participants considered PINs to provide "meta-security", and in many cases chose to use them to provide security on their mobile devices. They stated that a barrier, such as that provided by entering a challenge question response, improved perceptions of security, even in cases where the transparent authentication was running normally. This implies that the user looks to their understanding of traditional security to help understand new methods, which can be used to build acceptance for new security methods.

9.2.3 Perceptions of Traditional and Transparent Authentication

Consideration 4: Transparent authentication methods should not attempt to eschew or replace traditional security methods.

Users are familiar with provisions such as passwords and PINs, and have strongly-held beliefs as to the security provided (or lack thereof) by these methods. Similar to the previous design consideration, it may be the case that familiarity could help breed a sense of security. Furthermore, crafting a strong mental model of transparent authentication security may be easier if the device owner is first introduced to the explicit authentication method provided, then the transparent method.

The Framework includes a back-up security method that is based on explicit authentication methods. The chosen implementation of this is left to the application designer who uses the Framework, but challenge questions have been suggested as an option. Other options include passwords, PINs and explicit biometrics such as fingerprints. User familiarity with such provisions may bolster the user's mental model and help provide a feeling of enhanced security provision.

[1] An example of risk communication, although not from the security field, is the TV commercial that showed a frying egg with a voiceover stating "This is your brain on drugs".

> **Consideration 5:** Continuous, transparent authentication methods should respect the limitations of the platform for which they are designed.

This consideration takes into account the likelihood that the user is trying to achieve other tasks that may require security provisions, and not to "do security" itself. Thus, on a mobile device platform, considerations must be made for the bursty, frequent use pattern that characterizes this platform, as well as the limitations this platform has on memory, processor speed, and power consumption. These limitations are especially important when considering the continuous nature of the Framework presented in this research, since the frequency of recalculating device confidence may have an effect on these resources. Finally, the intended transparent nature of the Framework should also be kept in mind when selecting the biometrics to use in the Framework. They should be ones that may be gathered while the user goes about regular tasks, and the number should be sufficient to provide the accuracy needed by the specific application while respecting the resource limitations of mobile devices.

Efforts have been made in the Framework design to select processes that minimize complexity. For instance, the pattern classifiers tested are all simple to program and relatively fast in decision making. The processes and data structures that define the Framework have been selected to minimize battery use and memory. These choices were made deliberately, but it is left to future simulation work to determine whether these choices are as efficient as necessary. The choice of biometrics respects the platform since they can be gathered while the user goes about regular tasks. Finally, the continuous and transparent nature of the Framework blends well with the bursty nature of mobile device use since it largely removes the need for frequent explicit authentication.

9.3 Research Contributions

This research provided a framework for continuous, transparent authentication on mobile devices. The specific research question that drove this research was as follows:

> It is possible to verify the identity of the current user of a mobile device in a secure, continuous, transparent and passive manner by using a combination of behavioral biometrics. Such authentication will not normally require explicit owner action, but will instead rely on the owner's usual interaction with the mobile device. Finally, such a transparent authentication method will be acceptable to device owners.

This research question provided the basis for the hypotheses that drove the research and the creation of the Transparent Authentication Framework, as follows:

> **H1:** Behavioral biometrics such as keystroke dynamics and speaker verification are sufficiently distinctive to contribute to verification of the identity of a mobile device owner.

This hypothesis was addressed with the Keystroke Dynamics and Speaker Verification feasibility studies. In both cases, the studies showed that the owner of a mobile device had sufficient patterns in their typing and speaking patterns to justify using it as a behavioral biometric in the Transparent Authentication Framework. However, neither method provided low enough error rates to justify using it as a sole means of identity verification. These two feasibility studies have sufficiently low error rates to justify extended studies of the same type that have more participants and more data per participant. As such, this hypothesis is accepted since there is enough information in each of these biometrics to contribute to mobile device owner identity verification. Care should be taken, however, to select the behavioral biometrics used within the Framework to ensure that those chosen are sufficiently distinctive to make a similar contribution.

> **H2:** Combining keystroke dynamics and speaker verification into a multimodal behavioral biometric reduces the error rates seen with the individual biometrics.

This hypothesis was addressed with the Multimodal Biometrics feasibility study, in which the speaker verification and keystroke dynamics study results were combined in a measurable and repeatable way. The results of this study showed that both the Naïve and Posterior Probability Methods (NM and PPM, respectively) showed improvements in error rates via higher AUC levels. While these results were not statistically significant at all levels, they do provide support for a larger study that further examines the applicability of these two combination methods. The results of this study indicate that applications that require lower security levels would benefit from using the Naïve Method because it favors user convenience over resisting intruder access (i.e., allows fewer false negatives). The Posterior Probability method, on the other hand, is preferred for higher security applications since it favors blocking intruders over the inconvenience of asking legitimate owners to re-authenticate. Thus, since the error rates were indeed often lower for the multimodal biometrics compared to the single biometrics, this hypothesis is accepted.

> **H3:** It is possible to gather keystroke dynamics and speaker verification biometrics while the mobile device user goes about other tasks on the device.

This hypothesis was addressed by undertaking the keystroke dynamics and speaker verification feasibility studies. While it is conceptually possible to collect both keystroke data and

voice samples while the device owner goes about regular tasks on the device such as writing email or making phone calls, the implementation of this functionality is somewhat less possible. The Apple iPhone and iPod Touch environment was chosen for the experimental platform. While writing the mobile device applications that supported these two studies, it became clear that Apple did not allow processes to run in the background as a regular working condition. Since this is a requirement of gathering either keystroke dynamics or speaker verification biometrics in a transparent manner, further work into backgrounding these applications was not pursued. However, the Android development environment does allow for background processes, and thus further research into this platform may provide a different result for this hypothesis. This hypothesis is rejected for Apple mobile device environments since background processes are disallowed, but accepted in theory since it is demonstrably possible on the Android platform.

Another consideration that was discovered during this research was the apparent mismatch between the idea of keystroke dynamics and the reality of it. During the keystroke dynamics feasibility study, several participants indicated that they would not feel comfortable with participating in the study if their keystrokes were sampled in the background. These participants were happy to provide their keystroke patterns for the study as long as they could choose what to type. However, the participants in the Transparent Authentication Perceptions study did not seem to have an issue with their keystrokes being sampled and used for authentication purposes. One possible reason for this difference in opinion lies in the differences between the two studies' design. In the keystroke dynamics study, the participants knew that their patterns would be removed from their device and compared to other such patterns, but in the transparent authentication perceptions study, the participants were led to believe that their keystroke patterns remained on the device. This is a small but very important difference, since the former has privacy implications while the latter appears to protect privacy. The discovery, then, is that privacy and control over their distinguishing information and data is important to device owners, and therefore must be considered carefully during design of applications that are based on the Framework presented here.

> **H4:** Mobile device owners would consider using a transparent authentication method if it was available to them.

This hypothesis was addressed by the Transparent Authentication Perceptions (TAP) Study. The results of this study showed that the participants would at least consider using such an authentication method. There were several caveats to this claim, however. First, several participants indicated that they would trial such an application, but would not hesitate to remove or disable it if it either kept them from their data and device functionality, or if it seemed to not block unauthorized users (i.e., if their friend started using the device and

was allowed the same access as the owner). The idea of security as a barrier was a theme identified during the TAP study; the number and frequency of the barriers to intended task completion should be considered when using the Framework. This hypothesis is accepted.

9.3.1 Major Contributions

This research has provided novel contributions to the field of authentication, particularly in the mobile device environment. The major contributions, which extend the design considerations and hypotheses provided in the previous sections, are as follows:

1. Developed a framework for continuous, transparent authentication on mobile devices that is intended to be independent of both device type and model, and also of the operating system type and version.
2. Designed the Framework so that it is plausible on a mobile device without a dependence on offline processing in order to allow the owner's data to remain on the device and within their control.
3. Extended keystroke dynamics research into soft keyboards on mobile devices.
4. Gathered data on user perceptions of transparent authentication, and used these perceptions to inform the creation of the Transparent Authentication Framework, with the hope that this will help move transparent authentication beyond the research lab and into more regular use.
5. Provided support for the use of multimodal biometrics in such a Framework, which has not been proposed previously, and allowed for flexibility in the number and type of biometrics chosen.

9.3.2 Minor Contributions

In addition to the novel contributions discussed in the previous section, this research also provides support for the work of other researchers in the field. Although these minor contributions were not directly mandated by the research question and hypotheses that define this research, they are important since they may help advance the field of transparent authentication.

1. Supported the results of similar work in keystroke dynamics, including verifying the conclusion that key hold times are not very distinctive on mobile devices and showing similar error rates to other research in this area.

2. Provided methods of combining multiple biometrics that depend on the posterior probabilities provided by pattern classifiers, rather than on individual score matrices. One interpretation of this result is that using probabilities may prove to use less memory, which is a consideration in the mobile device platform.

3. Provided up-to-date research into the current mobile device security provisions used by device owners through the TAP study.

4. Provided a survey of current research in keystroke dynamics on mobile devices, desktop and laptop computers [133].

9.4 Future Work

This research has provided novel contributions to the field of authentication through the answers provided to the overarching research question. In providing these answers, however, more questions have been identified. These questions provide a rich source of future work in this field.

1. Extended studies on keystroke dynamics and speaker verification on mobile devices. Extended implies both more participants and larger datasets, as well as potential examination of alternative pattern classifiers to those studied in this research.

2. Creation of a simulation of the Transparent Authentication Framework. This will provide a method of verifying the Framework's inclusions, allow for stringent testing of assumptions such as the usefulness of identifying attackers quickly, and allow multiple biometrics scenarios to be tested free of the constraints of a mobile device platform. This future work has several objectives: to justify the Framework's parts, to provide a basis for a proof-of-concept application, to allow refinement of the Framework's parts and to identify any gaps in the Framework.

3. Creation of a proof-of-concept prototype application that is based on the Framework and that runs on a mobile device. The resulting application would be based on the results of the simulations, and would focus on its processor, memory, and power needs and usage. Once tested, the application may be used to conduct further tests into the Framework's usability and device owner requirements.

4. Experiments into whether typing and voice patterns (and potentially other biometrics) can be gathered in a transparent manner by using backgrounding on mobile device application platforms. This work may be conducted in unison with creating the proof-of-concept application. Examples include gait and implicit facial recognition.

5. Explore the relationship between tasks and data that have different required confidence levels. For example, if one low security and one medium security task each attempt to access data that has been assigned a high security level, should access to the data be granted? One option is to default to the highest level, thus disallowing data access, but this is a complex problem that should be examined, perhaps by using solutions in different fields that have similar problems.

9.5 Conclusions

This research provided a description of a framework for continuous, transparent authentication on mobile devices, called the Transparent Authentication Framework. To provide support for the use of behavioral biometrics within the Framework, feasibility studies into keystroke dynamics, speaker verification, and the multimodal combination of these biometrics were undertaken. To answer questions about the usefulness and needs of users who may use such a Framework, a study into user perceptions of transparent authentication was implemented.

The results of this research show that transparent authentication on mobile devices has potential both in terms of technology and support from device owners. Results of this research include support for keystroke dynamics, speaker verification and multimodal biometrics for use in the Transparent Authentication Framework, further understanding of user perceptions of transparent authentication in general, and specific suggestions for functionality that have been reflected in the Framework design.

Future work includes creating simulations of the processes and data structures that comprise the Framework, and a proof-of-concept implementation on a mobile device platform that supports application backgrounding. To support this future work, larger studies with more participants and more data into the usefulness of keystroke dynamics and speaker verification are justified by the feasibility study results reported in this research. Examination of other possible biometrics such as gait and implicit facial recognition may provide support for other biometrics as well. Finally, a large-scale user study of the utility and usability of the proof-of-concept implementation will identify ways in which the Framework may be improved.

Appendix A

Transparent Authentication Perceptions Study Interview Questions

The Transparent Authentication Perceptions Study received ethics approval from the College of Science and Engineering (formerly the Faculty of Information and Mathematical Sciences) on August 7, 2012 under the ethics number CSE01076. No changes to the experimental design described in the ethics application were required in order to gain ethics approval. The following pages show the questions asked during the semi-structured interview conducted during the study. Not all questions asked are represented here. The interviewer may have asked additional questions depending on the answers provided by the participant.

Subject ID: _____ Interview Date: _____ Interview Duration: _____

1. Were you able to complete all the tasks given to you? YES / NO
   (a) Notes:

2. Did you turn off the transparent authentication system (i.e., tapped the button marked "Turn Off Security")? YES / NO
   (a) Why or why not?

3. Did you use the challenge question feature? YES / NO
   (a) Why or why not?

4. Assume for a moment that you were placing each task from the study into a security level that you think is most appropriate given how you use your mobile device and how sensitive you think each task is. Use the 3-point Likert scale to assign each task from the study into what level you think it should be in.

   Read Document: Low (1) Medium (2) High (3)

   Take Photo: Low (1) Medium (2) High (3)

   Send Email: Low (1) Medium (2) High (3)

   View Photo: Low (1) Medium (2) High (3)

   Make Local Call: Low (1) Medium (2) High (3)

   Change Device PIN: Low (1) Medium (2) High (3)

   Make International Call: Low (1) Medium (2) High (3)

5. How many security level choices would you like to have? Is Low/Med/High accurate enough, or should there be more choices?

6. Use the 5-point Likert scale to indicate how easy or difficult it was to complete each of the tasks (1 is very easy, 5 is very difficult):

   Read Document: 1 2 3 4 5

   Take Photo: 1 2 3 4 5

   Send Email: 1 2 3 4 5

   View Photo: 1 2 3 4 5

   Make Local Call: 1 2 3 4 5

   Change Device PIN: 1 2 3 4 5

   Make International Call: 1 2 3 4 5

7. What did you like about using the transparent authentication system?

8. What did you dislike about using the transparent authentication system?

9. Would you use a transparent authentication method on your own mobile device? YES / NO
   (a) Why or why not?

10. Using the 5-point Likert scale, indicate how well protected you thought the data on the device was. 1 is very unprotected, 2 is somewhat unprotected, 3 is neither protected nor unprotected, 4 is somewhat protected and 5 is very protected.

(a) Why did you select this level?

11. What security mechanism do you currently use on your mobile device?

12. When compared to using your usual security mechanism as the sole security method on a mobile device, did you feel that using a transparent authentication method was more secure, less secure, or about the same? Use the Likert scale for this 1 is a lot less secure, 2 is somewhat less secure, 3 is about the same, 4 is somewhat more secure, and 5 is a lot more secure.

1 2 3 4 5

(a) Why?

13. When compared to using no security method at all on a mobile device, did you feel that using a transparent authentication method was more secure, less secure, or about the same? Use the Likert scale for this 1 is a lot less secure, 2 is somewhat less secure, 3 is about the same, 4 is somewhat more secure, and 5 is a lot more secure.

1 2 3 4 5

(a) Why?

Bibliography

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114 – ff., April 1965.

[2] Canalys, "Smart Phones Overtake Client PCs in 2011," Online: http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011. Last accessed: October 17, 2012.

[3] W. B. Glisson, T. Storer, G. Mayall, I. Moug, and G. Grispos, "Electronic Retention: What Does Your Mobile Phone Reveal About You?" *International Journal of Information Security*, vol. 10, no. 6, pp. 337 – 349, 2011.

[4] A. Adams and M. A. Sasse, "Users Are Not the Enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40 – 46, December 1999.

[5] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password Memorability and Security: Empirical Results," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 25–31, Sept. – Oct. 2004.

[6] M. A. Sasse, S. Brostoff, and D. Weirich, "Transforming the 'Weakest Link': A Human-Computer Interaction Approach to Usable and Effective Security," *BT Technical Journal*, vol. 19, no. 3, pp. 122 – 131, July 2001.

[7] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulous, R. Govindan, and D. Estrin, "Diversity in Smartphone Usage," in *Proceedings of the 8th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2010, pp. 179 – 194.

[8] H.-H. Jo, M. Karsai, J. Kertèsz, and K. Kaski, "Circadian Patterns and Burstiness in Mobile Phone Communication," *New Journal of Physics*, vol. 14, 2012.

[9] J. Bonneau and S. Preibusch, "The Password Thicket: Technical and Market Failures in Human Authentication on the Web," in *Proceedings of the Ninth Workshop on the Economics of Information Security (WEIS 2010)*, 2010.

[10] D. Weirich and M. A. Sasse, "Pretty Good Persuasion: A First Step Towards Effective Password Security in the Real World," in *Proceedings of the 2001 Workshop on New Security Paradigms (NSPW '01)*, 2001, pp. 137 – 143.

[11] D. Florêncio and C. Herley, "A Large-Scale Study of Web Password Habits," in *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, 2007, pp. 657 – 666.

[12] N. Clarke, S. Furnell, and P. Reynolds, "Biometric Authentication for Mobile Devices," in *Proceedings of the 3rd Australian Information Warfare and Security Conference 2002*, 2002, pp. 61 – 69.

[13] R. W. Frischholz and U. Dieckmann, "BioID: A Multimodal Biometric Identification System," *IEEE Computer*, vol. 33, no. 2, pp. 64 – 68, 2000.

[14] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: A Tool for Information Security," *IEEE Transactions on Information Forensics and Security*, vol. 1(2), pp. 125 – 143, 2006.

[15] S. Chiasson, P. C. van Oorschot, and R. Biddle, "Graphical Password Authentication Using Cued Click Points," in *Proceedings of the 2007 European Symposium on Research in Computer Security*, ser. Lecture Notes in Computer Science, vol. 4734/2007. Springer Berlin / Heidelberg, 2007, pp. 359 – 374.

[16] Paul Dunphy and Andreas P. Heiner and N. Asokan, "A Closer Look at Recognition-Based Graphical Passwords on Mobile Devices," in *Proceedings of the 6th Symposium on Usable Privacy and Security*, 2010, pp. 26 – 38.

[17] N. Clarke, S. Karatzouni, and S. Furnell, *Emerging Challenges for Security, Privacy and Trust*, ser. IFIP Advances in Information and Communication Technology. Springer Boston, 2009, vol. 297/2009, ch. Flexible and Transparent User Authentication for Mobile Devices, pp. 1 – 12.

[18] N. Clarke, S. Furnell, B. Lines, and P. Reynolds, "Subscriber Authentication for Mobile Phones Using Keystroke Dynamics," in *Proceedings of the Third International Network Conference (INC 2002)*, 2002, pp. 347 – 355.

[19] N. Clarke and S. Furnell, "Authenticating Mobile Phone Users Using Keystroke Analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 1 – 14, January 2007.

[20] S. Karatzouni and N. Clarke, *New Approaches for Security, Privacy and Trust in Complex Systems*. Springer Boston, 2007, vol. 232/2007, ch. Keystroke Analysis for Thumb-based Keyboards on Mobile Devices, pp. 253 – 263.

[21] R. Sandhu and P. Samarati, "Authentication, Access Control and Audit," *ACM Computing Surveys*, vol. 28, no. 1, pp. 241 – 243, 1996.

[22] K. Renaud, *Security and Usability: Designing Secure Systems That People Can Use*. O'Reilly, 2005, ch. 6: Evaluating Authentication Mechanisms, pp. 103–128.

[23] S. Karatzouni, S. Furnell, N. Clarke, and R. A. Botha, "Perceptions of User Authentication on Mobile Devices," in *Proceedings of the 2007 ISOneWorld Conference*, 2007, p. CD Proceedings.

[24] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-Factor Authentication: Somebody you Know," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*. ACM, 2006, pp. 168 – 178.

[25] D. E. Denning and P. F. MacDoran, "Location-Based Authentication: Grounding Cyberspace for Better Security," *Computer Fraud & Security*, vol. 1996, no. 2, pp. 12 – 16, 1996.

[26] J. E. Bardram, R. E. K. r, and M. O. Pedersen, "Context-Aware User Authentication – Supporting Proximity-Based Login in Pervasive Computing," in *Proceedings of UBICOMP 2003: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, vol. 2864/2003, 2003, pp. 107 – 123.

[27] S. N. Porter, "A Password Extension for Improved Human Factors," *Computers & Security*, vol. 1, no. 1, pp. 54 – 56, 1982.

[28] U. Topkara, M. J. Atallah, and M. Topkara, "Passwords Decay, Words Endure: Secure and Re-Usable Multiple Password Mnemonics," in *Proceedings of the ACM Symposium on Applied Computing*, 2007, pp. 292 – 299.

[29] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle, "Improving Text Passwords Through Persuasion," in *Proceedings of the 4th ACM Symposium on Usable Privacy and Security (SOUPS)*, 2008, pp. 1 – 12.

[30] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and Longitudinal Evaluation of a Graphical Password System," *International Journal of Human-Computer Studies*, vol. 63, no. 1–2, pp. 102 – 127, July 2005.

[31] R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication," in *Proceedings of the 9th USENIX Security Symposium*, 2000, pp. 4 – 4.

[32] K. Renaud and J. Maguire, "Armchair Authentication," in *Proceedings of the 23rd British HCI Group Annual Conference*. British Computer Society, 2009, pp. 388 – 397.

[33] J. Goldberg, J. Hagman, and V. Sazawal, "Doodling our Way to Better Authentication," in *Proceedings of Computer Human Interaction (CHI'02): Extended abstracts on human factors in computing systems*. ACM, 2002, pp. 868 – 869.

[34] K. Renaud and J. Ramsay, "Now What Was That Password Again? A More Flexible Way of Identifying and Authenticating our Seniors," *Behaviour & Information Technology Special Issue: Designing Computer Systems for and With Older Users*, vol. 26, no. 4, pp. 309 – 322, July 2007.

[35] S. Brostoff and M. A. Sasse, *People and Computers XIV – Usability or Else: Proceedings of HCI*. Springer-Verlag, 2000, ch. Are Passfaces More Usable than Passwords: A Field Trial Investigation.

[36] S. Chiasson, R. Biddle, and P. C. van Oorschot, "A Second Look at the Usability of Click-Based Graphical Passwords," in *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS '07)*. ACM, 2007, pp. 1 – 12.

[37] E. Stobert, A. Forget, S. Chiasson, P. van Oorschot, and R. Biddle, "Exploring Usability Effects of Increasing Security in Click-Based Graphical Passwords," in *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10)*, 2010, pp. 79 – 88.

[38] A. Pashalidis and C. J. Mitchell, "A Taxonomy of Single Sign-On Systems," in *Information Security and Privacy*, ser. Lecture Notes in Computer Science, vol. 2727/2003, 2003, pp. 249–264.

[39] N. Clarke, *Transparent User Authentication: Biometrics, RFID and Behavioural Profiling*, 1st ed. Springer, 2011.

[40] L. Bauer, L. Cranor, M. Reiter, and K. Vaniea, "Lessons Learned from the Deployment of a Smartphone-Based Access-Control System," in *Proceedings of the 2007 Symposium on Usable Privacy and Security (SOUPS)*, ser. ACM International Conference Proceedings Series, vol. 229, ACM. ACM, 2007, pp. 64 – 75.

[41] N. Clarke, S. Furnell, P. Rodwell, and P. Reynolds, "Acceptance of Subscriber Authentication for Mobile Telephony Devices," *Computers & Security*, vol. 21, no. 3, pp. 220 – 228, 2001.

[42] S. Kowalski and M. Goldstein, "Consumers' Awareness of, Attitudes Towards, and Adoption of Mobile Phone Security," in *Proceedings of the 20th International Symposium on Human Factors in Telecommunication*, 2006.

[43] R. A. Botha, S. Furnell, and N. Clarke, "From Desktop to Mobile: Examining the Security Experience," *Computers & Security*, vol. 28, no. 3-4, pp. 130 – 137, 2009.

[44] N. Clarke and S. Furnell, "Authentication of Users on Mobile Telephones - A Survey of Attitudes and Practices," *Computers & Security*, vol. 24, no. 7, pp. 519 – 527, October 2005.

[45] M. Vojnovic, "On Mobile User Behaviour Patterns," in *Proceedings of the International Zurich Seminar on Communications*, 2008, pp. 26 – 29.

[46] C. Herley, P. C. van Oorschot, and A. S. Patrick, "Passwords: If We're So Smart, Why Are We Still Using Them?" in *Proceedings of the 13th International Conference on Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, vol. 5628/2009, 2009, pp. 230 – 237.

[47] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge Attacks on Smartphone Touch Screens," in *Proceedings of WOOT '10*, 2010.

[48] "Apple Buys into Fingerprint Recognition with AuthenTec Deal," *Biometric Technology Today*, vol. 2012, no. 8, p. 1, 2012.

[49] Office of Communications, "Mobile Citizens, Mobile Consumers," http://stakeholders.ofcom.org.uk/consultations/msa08/, November 2008.

[50] R. Ghosh and M. Dekhil, "I, Me and My Phone: Identity and Personalization using Mobile Devices," Digital Printing and Imaging Laboratory, HP Laboratories, Technical Report HPL-2007-184, November 2007.

[51] A. Bottoni and G. Dini, "Improving Authentication of Remote Card Transactions with Mobile Personal Trusted Devices," *Computer Communications*, vol. 30, no. 8, pp. 1697 – 1712, June 2007.

[52] M. Tanviruzzaman, S. I. Ahamed, C. S. Hasan, and C. O'Brien, "ePet: When Cellular Phone Learns to Recognize its Owner," in *Proceedings of ACM Workshop on Assurable & Usable Security Configuration (SafeConfig)*. Collocated with the ACM Conference on Computer and Communications Security (CCS), November 2009, pp. 13 – 18.

[53] F. Stajano, "Pico: No More Passwords!" in *Proceedings of the 19th International Workshop on Security Protocols*, ser. Lecture Notes in Computer Science, vol. 7114/2011. Springer Berlin / Heidelberg, 2011, pp. 49 – 81.

[54] A. D. Frankel and M. Maheswaran, "Feasibility of a Socially Aware Authentication Scheme," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference*, 2009, pp. 1 – 6.

[55] H. Gamboa and A. Fred, "A Behavioural Biometric System Based on Human Computer Interaction," in *Proceedings of SPIE*, vol. 5404-36, 2004.

[56] S. Yazji, X. Chen, R. P. Dick, and P. Scheuermann, *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5585/2009, ch. Implicit User Re-authentication for Mobile Devices, pp. 325 – 339.

[57] A. Ross and A. K. Jain, "Multimodal Biometrics: An Overview," in *Proceedings of the 12th European Signal Processing Conference (EUSIPCO)*, September 2004, pp. 1221 – 1224.

[58] NIST, "Summary of NIST Standards for Biometric Accuracy, Tamper Resistance, and Interoperability," NIST, Tech. Rep., 2002.

[59] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14(1), January 2004, pp. 4 – 20.

[60] J. L. Wayman, A. K. Jain, D. Maltoni, and D. Maio, *Biometric Systems: Technology, Design and Performance Evaluation*. Springer, 2005, ch. An Introduction to Biometric Authentication Systems, pp. 1 – 20.

[61] S. Kung, M. Mak, and S. Lin, *Biometric Authentication A Machine Learning Approach*, T. Kailath, Ed. Prentice-Hall, 2005.

[62] J. Ashbourn, *Biometrics: Advanced Identity Verification: The Complete Guide*. Springer Berlin / Heidelberg, 2000.

[63] A. K. Jain, P. Flynn, and A. Ross, Eds., *Handbook of Biometrics*. Springer, 2008.

[64] M. Golfarelli, D. Maio, and D. Maltoni, "On the Error-reject Tradeoff in Biometric Verification Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 786 – 796, July 1997.

[65] R. V. Yampolskiy and V. Govindaraju, "Behavioural Biometrics: A Survey and Classification," *International Journal of Biometrics*, vol. 1, no. 1, pp. 81 – 113, 2008.

[66] S. Bleha, C. Silvinsky, and B. Hussien, "Computer-Access Security Systems Using Keystroke Dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1217 – 1222, December 1990.

[67] S. Cho, C. Han, D. H. Han, and H.-I. Kim, "Web based Keystroke Dynamics Identity Verification Using Neural Network," *Journal of Organizational Computing and Electronic Commerce*, vol. 10, no. 4, pp. 295 – 307, 2000.

[68] A. Buchoux and N. Clarke, "Deployment of Keystroke Analysis on a Smartphone," in *Proceedings of the 6th Australian Information Security Management Conference*, 2008, pp. 40 – 47.

[69] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds, "A Tutorial on Text-Independent Speaker Verification," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 430 – 451, 2004.

[70] Q. Li, B.-H. Juang, C.-H. Lee, Q. Zhou, and F. K. Soong, "Recent Advancements in Automatic Speaker Authentication," *IEEE Robotics & Automation Magazine*, pp. 24 – 34, March 1999.

[71] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A Study of Inter-Speaker Variability in Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980 – 988, July 2008.

[72] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the Applicability of Touchscreen Input as Behavioral Biometric for Continuous Authentication," *IEEE Transactions on Information Forensics and Security*, vol. to appear., 2012.

[73] R. V. Yampolskiy, "Behavioral Modeling: An Overview," *American Journal of Applied Sciences*, vol. 5, no. 5, pp. 496 – 503, 2008.

[74] J. Fierrez-Aguilar, J. Ortega-Garcia, J. Gonzalez-Rodriguez, and J. Bigun, "Discriminative Multimodal Biometric Authentication Based on Quality Measures," *Pattern Recognition*, vol. 38, no. 5, pp. 777 – 779, May 2004.

[75] F. LeClerc and R. Plamondon, "Automatic Signature Verification: The State of the Art - 1989–1993," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 3, 1994.

[76] D. Cunado, M. S. Nixon, and J. N. Carter, "Using Gait as a Biometric, via Phase-Weighted Magnitude Spectra," in *Audio- and Video-based Biometric Person Authentication*, vol. 1206/1997, 1997, pp. 93 – 102.

[77] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Allisto, "Identifying Users of Portable Devices from Gait Pattern with Accelerometers," in *Proceedings*

of the IEEE International Conference on Acousics, Speech, and Signal Processing, vol. 2, 2005, pp. 973 – 976.

[78] D. Gafurov, E. Snekkenes, and T. E. Buvarp, "Robustness of Biometric Gait Authentication Against Impersonation Attacks," in *Proceedings of the Workshop on On the Move to Meaningful Internet Systems*, ser. Lecture Notes in Computer Science, vol. 4277/2006, 2006, pp. 479 – 488.

[79] O. Mazhelis and S. Puuronen, "A Framework for Behavior-Based Detection of User Substitution in a Mobile Context," *Computers & Security*, vol. 26, no. 2, pp. 154 – 176, 2007.

[80] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit Authentication through Learning User Behavior," in *Information Security*, ser. Lecture Notes in Computer Science, M. Burmester, G. Tsudik, and S. Magliveras, Eds. Springer Berlin / Heidelberg, 2011, vol. 6531, pp. 99–113.

[81] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric Personal Authentication Using Keystroke Dynamics: A Review," *Applied Soft Computing*, vol. 11, no. 2, pp. 1565 – 1573, March 2011.

[82] S. Hwang, S. Cho, and S. Park, "Keystroke Dynamics-Based Authentication for Mobile Devices," *Computers & Security*, vol. 28, no. 1-2, pp. 85 – 93, 2009.

[83] J.-F. Bonastre, F. Bimbot, L.-J. Boe, J. P. Cambell, D. A. Reynolds, and I. Magrin-Chagnolleau, "Person Authentication by Voice: A Need for Caution," in *Proceedings of Eurospeech 2003*, 2003.

[84] R. V. Yampolskiy, "Action-based user authentication," *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 3, pp. 281 – 300, 2008.

[85] J. Koreman, A. C. Morris, D. Wu, S. Jassim, H. Sellahewa, J. Ehlers, G. Chollet, G. Aversano, H. Bredin, S. Garcia-Salicetti, L. Allano, B. Ly-Van, and B. Dorizzi, "Multi-modal Biometric Authentication on the SecurePhone PDA," in *Proceedings of the 2nd International Workshop on Multimodal User Authentication*, 2006.

[86] R. R. Roberts, R. A. Maxion, K. S. Killourhy, and F. Arshad, "User Discrimination Through Structured Writing on PDAs," in *Proceedings of the International Conference on Dependable Systems & Networks*, 2007, pp. 378 – 387.

[87] S. N. Patel, J. S. Pierce, and G. D. Abowd, "A Gesture-Based Authentication Scheme for Untrusted Public Terminals," in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, 2004, pp. 157 – 160.

[88] Y. Nakkabi, I. Traore, and A. Ahmed, "Improving Mouse Dynamics Biometric Performance Using Variance Reduction via Extractors With Separate Features," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 6, pp. 1345 – 1353, November 2010.

[89] C.-J. Tsai, T.-Y. Chang, and Y.-J. Yang, "An Approach for User Authentication on Non-Keyboard Devices Using Mouse Click Characteristics and Statistical-Based Classification," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 11, pp. 7875 – 7886, November 2012.

[90] N. Zheng, A. Paloski, and H. Wang, "An Efficient User Verification System Via Mouse Movements," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 2011, pp. 139 – 150.

[91] Hannu Verkasalo, "Analysis of Smartphone User Behavior," in *Mobile Business and 2010 Ninth Global Mobility Roundtable*, 2010, pp. 258 – 263.

[92] R. Spillane, "Keyboard Apparatus for Personal Identification," IBM Technical Disclosure Bulletin, Tech. Rep. 17, 1975.

[93] J. Garcia, "Personal identification apparatus," U.S. Patent Number 4,621,334, November 1986.

[94] J. Young and R. Hammond, "Method and apparatus for verifying an individual's identity," U.S. Patent Number 4,805,222, February 1989.

[95] A. A. E. Ahmed, I. Traore, and A. Almulhem, "Digital Fingerprinting Based on Keystroke Dynamics," in *Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance (HAISA 2008)*, Plymouth, UK, July 2008, pp. 94 – 104.

[96] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke Dynamics with Low Constraints SVM Based Passphrase Enrollment," in *Proceedings of the 3rd International Conference on Biometrics: Theory, Applications, and Systems*, 2009, pp. 1 – 6.

[97] D. Gunetti and C. Picardi, "Keystroke Analysis of Free Text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312 – 347, August 2005.

[98] D. Hosseinzadeh and S. Krishnan, "Gaussian Mixture Modeling of Keystroke Patterns for Biometric Applications," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, no. 6, pp. 816 – 826, November 2008.

[99] R. Janakiraman and T. Sim, *Advances in Biometrics*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4642/2007, ch. Keystroke Dynamics in a General Setting, pp. 584 – 593.

[100] R. Joyce and G. Gupta, "Identity Authentication Based on Keystroke Latencies," *Communications of the ACM*, vol. 33, no. 2, pp. 168 – 176, February 1990.

[101] J. Leggett and G. Williams, "Verifying Identity via Keystroke Characteristics," *International Journal of Man-Machine Studies*, vol. 28, no. 1, pp. 67 – 76, January 1988.

[102] R. Maxion and K. S. Killourhy, "Keystroke Biometrics with Number-Pad Input," in *Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems and Networks*, 2010, pp. 201 – 210.

[103] K. Zhang and X. Wang, "Peeping Tom in the Neighborhood: Keystroke Eavesdropping on Multi-User Systems," in *Proceedings of the 18th USENIX Security Symposium*, 2009, pp. 17 – 32.

[104] P. Campisi, E. Maiorana, M. L. Bosco, and A. Neri, "User Authentication Using Keystroke Dynamics for Cellular Phones," *IET Signal Processing - Special Issue on Biometric Recognition*, vol. 3, no. 4, pp. 333 – 341, 2009.

[105] N. Clarke, S. Furnell, B. Lines, and P. Reynolds, "Keystroke Dynamics on a Mobile Handset: A Feasibility Study," *Information Management & Computer Security*, vol. 11, no. 4, pp. 161 – 166, 2003.

[106] H. Saevanee and P. Bhattarakosol, "Authenticating User Using Keystroke Dynamics and Finger Pressure," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference*, 2009, pp. 1 – 2.

[107] X. Huang, G. Lund, and A. Sapeluk, "Development of a Typing Behaviour Recognition Mechanism on Android," in *Proceedings of 11th IEEE International Conference on Trust, Security, and Privacy in Computing and Communications*, 2012, pp. 1342 – 1347.

[108] J. Mantyjarvi, J. Koivumaki, and P. Vuori, "Keystroke Recognition for Virtual Keyboards," in *Proceedings of the 2002 IEEE Conference on Multimedia and Expo*, vol. 2, 2002, pp. 429 – 432.

[109] S. Zahid, M. Shahzad, S. A. Khayam, and M. Farooq, *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5758, ch. Keystroke-Based User Identification on Smart Phones, pp. 224 – 243.

[110] S. Modi and S. J. Elliott, "Keystroke Dynamics Verification Using a Spontaneously Generated Password," in *Proceedings of 40th Annual IEEE International Carnahan Conferences Security Technology*, 2006, pp. 116 – 121.

[111] F. Monrose, M. Reiter, and S. Wetzel, "Password Hardening Based on Keystroke Dynamics," *International Journal of Information Security*, vol. 1, no. 2, pp. 69 – 83, February 2002.

[112] J. Robinson, V. Liang, J. Chambers, and C. MacKenzie, "Computer User Verification using Login String Keystroke Dynamics," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 2, pp. 236 – 241, March 1998.

[113] E. Maiorana, P. Campisi, N. Gonzalez-Carballo, and A. Neri, "Keystroke Dynamics Authentication for Mobile Phones," in *Proceedings of the 2011 Symposium on Applied Computing (SAC '11)*, 2011, pp. 21 – 26.

[114] S. Haidar, A. Abbas, and A. K. Zaidi, "A Multi-Technique Approach for User Identification through Keystroke Dynamics," in *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, 2000, pp. 1336 – 1341.

[115] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of Telephone Conversations Using Factor Analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059 – 1070, December 2010.

[116] S. E. Tranter and D. A. Reynolds, "An Overview of Automatic Speaker Diarization Systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557 – 1565, 2006.

[117] R. Brunelli and D. Falavigna, "Person Identification Using Multiple Cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 955 – 966, October 1995.

[118] H. Gish and M. Schmidt, "Text-Independent Speaker Identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 18 – 32, October 1994.

[119] T. Kinnunen and H. Li, "An Overview of Text-Independent Speaker Recognition: From Features to Supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12 – 40, January 2010.

[120] L. O'Gorman, L. Brotman, and M. Sammon, "Comparing Authentication Protocols for Securely Accessing Systems by Voice," in *Proceedings of the 2nd Secure Knowledge Management Conference*, September 2006.

[121] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," in *Proceedings of the 1992 IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP '92)*, vol. 1, 1992, pp. 517 – 520.

[122] M. Indovina, U. Uludag, R. Snelick, A. Mink, and A. K. Jain, "Multimodal Biometric Authentication Methods: A COTS Approach," in *Proceedings of Workshop on Multimodal User Authentication*, December 2003, pp. 99 – 106.

[123] A. Ross and A. K. Jain, "Information Fusion in Biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115 – 2125, September 2003.

[124] Y. A. Zuev and S. Ivanov, "The voting as a Way to Increase the Decision Reliability," *Journal of the Franklin Institute*, vol. 336, no. 2, pp. 361 – 378, March 1999.

[125] K. Iwano, T. Hirose, E. Kamibayashi, and S. Furui, "Audio-Visual Person Authentication Using Speech and Ear Images," in *Proceedings of Workshop on Multimodal User Authentication*, 2003, pp. 85 – 90.

[126] J. Rokita, A. Krzyzak, and C. Suen, *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5112, ch. Cell Phones Personal Authentication Systems Using Multimodal Biometrics, pp. 1013 – 1022.

[127] L. Allano, A. C. Morris, H. Sellahewa, S. Garcia-Salicetti, J. Koreman, S. Jassim, B. Ly-Van, D. Wu, and B. Dorizzi, "Non-Intrusive Multi-Biometrics on a Mobile Device: A Comparison of Fusion Techniques," in *Proceedings of the SPIE Conference on Biometric Technology for Human Identification III*, 2006.

[128] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, "Fusion of Face and Speech Data for Person Identity Verification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1065 – 1074, 1999.

[129] B. Duc, E. S. Bigun, J. Bigun, G. Maitre, and S. Fischer, "Fusion of Audio and Video Information for Multi Modal Person Authentication," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 835 – 843, September 1997.

[130] S. Garcia-Salicetti, C. Beumier, G. Chollet, B. Dorizzi, J. L. les Jardins, J. Lunter, Y. Ni, and D. Petrovska-Delacretaz, *Audio- and Video-based Biometric Person Authentication*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, vol. 2688/2003, ch. BIOMET: A Multimodal Person Authentication Database Including Face, Voice, Fingerprint, Hand and Signature Modalities, p. 1056.

[131] B. C. Kovoor, S. M.H., and K. P. Jacob, "A Prototype for a Multimodal Biometric Security System based on Face and Audio Signatures," *International Journal of Computer Science and Communication*, vol. 2, no. 1, pp. 143 – 147, January - June 2011.

[132] N. Poh and J. Korczak, *Audio- and Video-based Biometric Person Authentication*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 2901/2001, ch. Hybrid Biometric Person Authentication Using Face and Voice Features, pp. 348 – 353.

[133] H. Crawford, "Keystroke Dynamics: Characteristics and Opportunities," in *Proceedings of the 8th Annual Conference on Privacy, Security, and Trust (PST)*, 2010, pp. 205 – 212.

[134] J. H. Holmes, "Quantitative Methods for Evaluating Learning Classifier System Performance in Forced Two-Choice Decision Tasks," in *Proceedings of the 2nd International Conference on Learning Classifier Systems (IWLCS99)*. Morgan Kaufmann Publishers Inc., 1999.

[135] S. Bengio and J. Mariéthoz, "A Statistical Significance Test for Person Authentication," in *Proceedings of the Speaker and Language Recognition Workshop*, 2004, pp. 237 – 244.

[136] K. L. Adair, S. T. Parthasaradhi, and J. Kennedy, "Real World Evaluation: Avoiding Pitfalls of Fingerprint System Deployment," Lumidigm, Whitepaper, 2008.

[137] C. X. Ling, J. Huang, and H. Zhang, "AUC: A Statistically Consistent and More Discriminating Measure than Accuracy," in *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2003, pp. 519 – 526.

[138] "European Standard for Access Control Systems," BSI British Standards Publications, Tech. Rep. EN 50133-1, 1997.

[139] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit Authentication for Mobile Devices," in *Proceedings of the 4th USENIX Conference on Hot Topics in Security (HotSec '09)*, 2009, pp. 9–9.

[140] M. D. Corner and B. D. Noble, "Zero-interaction Authentication," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networks*, 2002, pp. 1 – 11.

[141] P. Briggs and P. L. Olivier, "Biometric Daemons: Authentication Via Electronic Pets," in *Proceedings of Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 2423 – 2432.

[142] P. Pullman, *Northern Lights*. Scholastic, 2007.

[143] S. Marsh and P. Briggs, "Defining and Investigating Device Comfort," in *Proceedings of the Fourth IFIP WG 11.11 International Conference on Trust Management*, 2010, pp. 17–24.

[144] ——, *Computing with Social Trust*. Springer, 2009, vol. 1, ch. Examining Trust, Forgiveness and Regret as Computational Concepts, pp. 9 – 43.

[145] S. Marsh and M. Dibben, "Trust, Untrust, Distrust, and Mistrust; an Exploration of the Dark(er) Side," in *Proceedings of iTrust 2005*, ser. Lecture Notes in Computer Science, vol. 3477, 2005.

[146] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulous, "Progressive Authentication: Deciding When to Authenticate on Mobile Phones," in *Proceedings of USENIX Security Symposium '12*, 2012, p. to appear.

[147] J. M. Allen, L. A. McFarlin, and T. Green, "An In-Depth Look into the Text Entry User Experience on the iPhone," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 52(5), 2008, pp. 508 – 512.

[148] M. Conti, I. Zachia-Zlatea, and B. Crispo, "Mind How You Answer Me!: Transparently Authenticating the User of a Smartphone when Answering or Placing a Call," in *Proceedings of the 6th ACM Symposium on Information, Computer, and Communications Security*, 2011, pp. 249 – 259.

[149] S. Furnell, N. Clarke, and S. Karatzouni, "Beyond the PIN: Enhancing user authentication for mobile devices," *Computer Fraud & Security*, vol. 8, pp. 12 – 17, 2008.

[150] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, 2001.

[151] M. Becher, F. Freiling, J. Hoffman, T. Holz, S. Uellenbeck, and C. Wolf, "Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices," in *Proceedings of 2011 IEEE Symposium on Security & Privacy*, 2011, pp. 96 – 111.

[152] A. Field and G. Hole, *How to Design and Report Experiments*. SAGE Publications, 2008.

[153] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, 1995, pp. 115 – 121.

[154] Y. Bengio and Y. Grandvalet, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation," *Journal of Machine Learning Research*, vol. 5, pp. 1089 – 1105, 2005.

[155] A. P. Bradley, "The Use of the Area Under the ROC curve in the Evaluation of Machine Learning Algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145 – 1159, July 1997.

[156] K. S. Killourhy and R. A. Maxion, "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009, pp. 125 – 134.

[157] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text," in *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pp. 69 – 71.

[158] A. Canavan, D. Graff, and G. Zipperlen, "CALLHOME American English Speech," Linguistic Data Consortium, Philadelphia, Tech. Rep., 1997.

[159] A. Canavan and G. Zipperlen, "CALLFRIEND American English Non-Southern Dialect," Linguistic Data Consortium, Philadelphia, Tech. Rep., 1996.

[160] R. H. Woo, A. Park, and T. J. Hazen, "The MIT Mobile Device Speaker Verification Corpus: Data Collection and Preliminary Experiments," in *IEEE Workshop on Speaker and Language Recognition*, 2006, pp. 1 – 6.

[161] ALIZE, "ALIZE: Open Source Platform For Biometrics Authentification," Online, http://mistral.univ-avignon.fr/index.html, 2012.

[162] G. Gravier, *SPro Speech Signal Processing Toolkit, release 4.0*, INRIA France, August 2003.

[163] ——, "Spro project, http://spro.gforge.inria.fr/," 2010.

[164] H. Beigi, *Fundamentals of Speaker Recognition*. Springer, 2011.

[165] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumochel, "Speaker and Session Variability in GMM-Based Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1448 – 1460, May 2007.

[166] R. Snelick, M. Indovina, J. Yen, and A. Mink, "Multimodal Biometrics: Issues in Design and Testing," in *Proceedings of the 5th International Conference on Multimodal Interfaces*, 2003, pp. 68 – 72.

[167] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Fusion Strategies in Multimodal Biometric Verification," in *Proceedings of the 2003 International Conference on Multimedia*, vol. 3, 2003, pp. 5 – 8.

[168] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226 – 239, March 1998.

[169] T. Murakami and K. Takahashi, "Accuracy Improvement with High Convenience in Biometric Indentification Using Multihypothesis Sequential Probability Ratio Test," in *Proceedings of the 1st IEEE International Workshop on Information Forensics and Security*, 2009, pp. 66 – 70.

[170] C. A. Bailer-Jones and K. Smith, "Combining Probabilities," Max Planck Institute for Astronomy, Heidelberg, Tech. Rep. GAIA-C8-TN-MPIA-CBJ-053, 2011.

[171] J. F. Kelley, "An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications," *ACM Transactions on Office Information Systems*, vol. 2, no. 1, pp. 26 – 41, March 1984.

[172] A. Strauss and J. M. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd ed. SAGE Publications, 1998.

[173] V. Bontchev, "Virusability of Modern Mobile Environments," in *Proceedings of Virus Bulletin 2007*, 2007.

[174] Q. Xiao, "Security Issues in Biometric Authentication," in *Proceedings of the 6th Annual IEEE Information Assurance Workshop*, 2005, pp. 8 – 13.

[175] K. Mitnick and W. Simon, *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, 2002.

[176] B. Geller, J. Almog, P. Margot, and E. Springer, "A Chronological Review of Fingerprint Forgery," *Journal of Forensic Sciences*, vol. 44, no. 5, pp. 963 – 969, 1999.

[177] W. W. Harper, "Fingerprint "Forgery". Transferred Latent Fingerprints," *Journal of Criminal Law and Criminology*, vol. 28, no. 4, pp. 573 – 580, 1937.

[178] J. Galbally-Herrero, J. Fierrez-Aguilar, J. Rodriguez-Gonzales, F. Alonso-Fernandez, J. Ortega-Garcia, and M. Tapiador, "On the Vulnerability of Fingerpring Verification Systems to Fake Fingerprints Attacks," in *Proceedings of the 40th Annual IEEE International Carnahan Conferences Security Technology*, 2006, pp. 130 – 136.

[179] K. S. Killourhy and R. A. Maxion, "Why Did My Detector Do *That*?! Predicting Keystroke-Dynamics Error Rates," in *Proceedings of 13th International Symposium on Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, vol. 6307. Springer-Verlag Berlin Heidelberg, 2010, pp. 256 – 276.

[180] J. N. Bailenson, N. Yee, K. Patel, and A. C. Beall, "Detecting Digital Chameleons," *Computers in Human Behavior*, vol. 24, no. 1, pp. 66 – 87, January 2008.

[181] F. Asgharpour, D. Liu, and L. J. Camp, "Mental Models of Security Risks," in *Proceedings of Financial Cryptography and Data Security 2007*, ser. Lecture Notes in Computer Science, vol. 4886/2007, 2007, pp. 367 – 377.

[182] L. J. Camp, "Mental Models of Privacy and Security," *IEEE Technology and Society Magazine*, vol. 28, no. 3, pp. 37 – 46, 2009.