Implicit Authentication On Mobile Devices

Tobias Stockinger

Abstract— One major disadvantage of mobile devices is their liability to theft. Since people make more intense use of their smart phones to browse the web and store potentially sensitive data, it is proposed to use additional measures to secure one's device. Implicit authentication for mobile devices is a promising solution to the problem. Analyzing user behavior and biometric characteristics can be used as additional measure to verify the user of a handset. Thus, sensors that are already inside contemporary smart phones are efficiently utilized for an area, that they were not originally designed for. Not only security is increased through implicit authentication techniques, but also usability. Inconvenient password requests by a phone can be reduced to a minimum. However, there are certain challenges in finding solutions to implement continuous, implicit authentication. Among these are the limited battery capacity and computational resources. Smart phones do not offer as much CPU power as cloud computing, so it is suggested to make use of these external resources, that are even dynamically adjustable.

Index Terms—Security, Smartphone, Implicit Authentication, Usability, Mobile Devices, Privacy

1 Introduction

The main purpose of authentication is to ensure that only the rightful owner of a certain device, account or document is granted access to it. In other words, this thought includes sealing the device off from possible adversaries such as thieves, impostors or also curious friends. However, current authentication fails in certain cases where an adversary somehow found out the credentials, e.g. Personal Identification Numbers (PIN) or any password.

One can imagine an everyday scenario: Alice's smart phone was stolen while she was at university. In the auditorium, Bob could have seen her typing in the PIN while he was sitting behind her ("Shoulder Surfing" [30]). As soon as Alice remarks the missing phone, she is likely to take measures to have the SIM card locked and request a remote deactivation of most phone functionality to protect her data (which is possible on newer phones [3]). Until that lock is established, Bob - the thief - has access to sensitive information and can even cause financial as well as reputational damage to Alice.

Although it is harder for an attacker to access and use the phone without knowing the PIN, many people do not secure their devices effectively with this type of authentication: PINs are usually only required at startup. After successfully verifying the identity of the user, the phone is henceforth unlocked. A lot of phones offer to re-authenticate after recovering from stand-by, but users seem to refrain from PINs/Passwords/Passcodes for usability reasons (or even renounce the PIN entirely) [7], [17]. Mobile devices complicate password entry because they have smaller keyboards. As a consequence, users are motivated to opt for simpler, therefore weaker, passwords which do not withstand common attacks [21].

There are a number of reasons, why securing personal data is especially important on mobile devices. For example, if the smart phone's email application is accessible for anyone, one part of the most sensitive data is vulnerable without much effort. Some emails include passwords for specific accounts, where an attacker can then log into. Furthermore, the majority of on-line services only requires a valid email address to reset the password. The impostor can reset many passwords for different accounts entailing a complete exclusion of the legitimate user. This weighs heavy, notably for financial accounts, such as PayPal [22].

In this paper, the principle of *implicit authentication* is presented, which tries to thwart the above-noted scenario and protect the user's data from misuse. Its main idea is to utilize user biometrics or behavioral characteristics to verify the rightful owner of a device in addition to traditional authentication like passwords.

The rest of the paper is structured as follows: *Section 2* states some necessary definitions before a short overview of authentication methods is given in *section 3*. Hereafter, the functional principles of implicit authentication in general (*section 4*) and for smart phones in particular (*section 5*) are explained. A summary of adversary characteristics is presented in *section 6*. The paper concludes by showing the limitations of implicit authentication (*section 7*) and a look into possible future development (*section 8*).

- *Tobias Stockinger is studying Media Informatics at the University of Munich, Germany, E-mail: stockinger@cip.ifi.lmu.de*
- *This research paper was written for the Media Informatics Advanced Seminar on Ubiquitous Computing, 2011*

2 Definitions

When talking about authentication, it is helpful to give some basic definitions regarding this matter. The process of *authentication*, or also *verification*, validates a claimed identity by matching it to a known set of identities [5]. In other words authentication answers the question *"Am I who I claim to be?"* [13]. The result of the one-to-one test has a binary output: The answer can be `true` or `false`. However, there are certain degrees, i.e. thresholds, of deciding whether the identity can be confirmed. Mostly, this happens through the calculation of a one-to-one matching *score* (see *section 4.3.3*) [5].

In contrast, *identification* has a different purpose. When *identifying* a person, that person does not his or her identity [5]. Rather, the system has to find out itself who is interacting, through matching certain characteristics of a client to models in the database. This is accomplished in a one-to-many matching process. As a consequence, *identification* searches for an answer to the question *"Who am I?"*[13]. It is then assumed, that the dataset with the highest similarity represents the individual. Thus, identification returns a vector or tuple closest to the person's characteristics instead of binary answers.

3 Authentication Methods

After defining the most important terms, some of the most common features that can be utilized to perform authentication are explained. This section covers the general authentication area and is not necessarily limited on mobile devices, which are presented in *section 5*, but it still provides the basis for understanding the specific conditions for portable devices. Note that *identification* does rely on some of the following cues as well. However, the focus of this paper remains on authentication, so identification is not treated in detail. Also, there is no distinction between implicit and explicit authentication just yet.

3.1 Passcodes

The most common way to authenticate a user is to explicitly ask her for a certain passcode [13]. The term "passcode" includes Personal Indentification Numbers (PINs), alphanumeric passwords and other graphical passwords [30]. This authentication approach relies on the

human brain's capability of remembering a preferably unique combination of numbers, letters or symbols. However it suffers from certain disadvantages:

(1) Even though people can remember a certain amount of different passcodes, the maximum capacity still seems limited. In 2006 it was found that a "heavy" user has an average of 21 passwords to remember [13]. As Gafurov et al. do not further specify the term "heavy" user, it is implied that a normal user is likely to have less than 21 passwords to bear in mind. This is either because the number of accounts is smaller compared to "heavy" users, or because a normal user is inclined to re-use a certain password for multiple accounts. This is aggravated by the fact that people usually are not motivated enough to linger over security issues [29]. Instead, about 81% of the users choose common words as their passwords, which are more susceptible to dictionary attacks [15]. (2) Another problematic issue are malware and other attacking methods. Even the strongest password can be compromised if a computer or mobile device is infected with keyloggers. Observing people while they type in their password is often referred to as shoulder surfing [30].

Despite their shortcomings, passcodes possess a huge advantage in comparison to biometrics: They are changeable. Once a password is compromised, it can easily be reset whereas for example face-, iris- or fingerprint-authentication lack this feature.

3.2 Tokens

Especially in business environments, the use of a special hardware device for a second factor authentication has established [26]. In case of SecurID the hardware token is a rather small device which displays a randomized number that is used as a second-level password [24]. Furthermore, this password is changed every 60 seconds which is supposed to make it even more secure. If used in combination with another portable device, this authentication method shows its strengths and weaknesses: On the one hand it is almost perfectly secure because the session timeout on the device that one tries to secure can be very low or even adapted to the presence of the device. If the token is not in the vicinity of the device, the session can be terminated, making it impossible for an attacker to access sensitive data. On the other hand, the token is as easily stolen or lost as the actual device. Moreover, this method requires a costly, highly evolved wireless network infrastructure to repeatedly send passwords to the token.

3.3 Biometric Cues

As we have seen in the previous paragraphs, passwords and tokens have some disadvantages, so that one should examine other authentication features. This section summarizes the different (physiological) biometric cues that can be used for authentication.

3.3.1 Face and Iris

Person identification through face recognition can be seen as the most intuitive way, because humans themselves are highly dependent on visual cues when it comes to recognizing someone. This approach already delivered working authentication systems in the early 1970s [6]. The general principle hereby is to capture a person's face through a camera. After digitizing the image, the pixels of certain regions of the face are compared to images that have already been acquired. Iris recognition operates a similar way, but only captures a high-resolution image from a person's eyes.

However, just like any other method this authentication scheme shows certain disadvantages: If the person's image isn't taken while in a similar position (e.g. because the camera angle is different), the matching faces difficulties. Moreover, people's faces change over time as people age, so it is suggested to implement an adaptive system.

Algorithms in this field are for example the Eigenfaces approach or the Fisherfaces algorithm [11]. The Fisherface-algorithm which is based on linear discriminant analysis has been found to produce low error rates, compared to the Eigenface approach [4]. Especially in different lighting situations "Fisherfaces" hold the upper hand.

3.3.2 Fingerprint

Fingerprint matching takes place by comparing minutiae (ridges and furrows on the finger), which are acquired using a fingerprint reading sensor [28]. This type of biometric authentication has become one of the most popular and is applicable to smart phones due to the rather small size of the sensor: A few solutions for business customers as well as consumer products already exist, such as the Motorla ATRIX 4G. Like almost any authentication method, it is possible to spoof a fingerprint. However, it takes a lot effort to successfully obtain and forge a person's fingerprint. A picture of a contemporary fingerprint reader on a smart phone is shown in *figure 1*.

Fig. 1. Fingerprint reader on the Motorola ATRIX 4G [20]

3.3.3 Voice / Speech

While speech recognition is to a high extent aimed at recognizing words and phrases independent of the speaker, voice- or speaker-recognition focuses on finding out *who* is speaking [6]. Voice recognition analyzes the acoustic characteristics of a speaker, such as pitch and phonetics. Especially in security applications, it is desirable to challenge a user with different phrases for each verification, so that the system is more robust against prerecorded samples. Voice recognition is more difficult or even impossible, if the user is sick or sore. Furthermore, the voice of a certain person might sound different in the morning than it does in the evening due to stressing during the day, e.g. singing or loud talking [11]. Additional background noise in loud environments might hamper any voice recognition.

3.3.4 Other Biometric Cues

Beside the above presented biometrics, a few other methods can be found in the literature, which shall not be explained in detail here. Handwritten signature comparison is usually done by visual inspection. That means a person usually checks another person's signature. However this authentication method also shows some potential to be exploited for automatic authentication [16]. Other possible authentication methods are based on electroencephalograms (EEG) and electrocardiograms (ECG) [11]. Therein special sensors measure electrical activity on the head (EEG) or around the Heart (ECG). Their major advantage is the fact, that they do not exclude any human being, since everyone has a heartbeat and a brain. However, one sensor does not suffice, which causes usability issues.

3.4 Behavioral Cues

Some papers in the area of biometrics consider behavior *biometric* as well. However, throughout this paper biometrics are seen as pure physiologic attributes. For example, someone can't immediately change his or her fingerprint whereas typing speed or gait can be manipulated on command.

3.4.1 Gait

First approaches of analyzing the individual characteristics of human gait were found in the 1970s [10]. Since then, researchers have tried to automatically identify people from their walk. After finding ways

of visual gait recognition, technology has evolved and gait data can now be collected using acceleration sensors [1]. An image of gait patterns, is shown in *figure 2*, where one can see two examples of gait cycles. On the left, the graph rises because the persons carrying the accelerometer were standing still at the beginning. After a short timespan both graphs show characteristic slopes which represent steps or gait cycles. Examining the two different plots, one can notice that the slopes differ in duration, for example. That examination can take place automatically, which is the main idea of gait recognition.

Due to the fact, that modern smart phones have already built-in gyroscopes, this method is applicable for mobile devices without hardware changes. However, one must consider, that there are a lot of things that can alter gait. Just to name a few: Footwear, ground surface, carrying load and injuries are likely to create notable impact on a person's walking style [13]. Furthermore, gait recognition fails if the user does not walk at all, because she has been seated for some time.

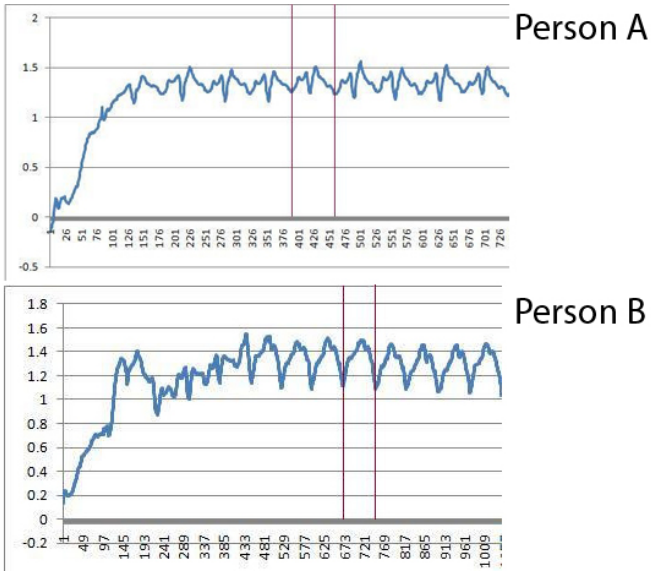

Fig. 2. Gait patterns (combined for x,y,z axes) gained from two different persons [27]

3.4.2 Keystroke

First keystroke authentication algorithms were found in the late 1980s by Card et al. [8]. They measured and analyzed the time span between two keystrokes, which they termed *diagraph*. Ever since then, this approach has been enriched with other factors such as key hold time and error rates [21]. In order to provide more reliable scores, keystroke based authentication usually requires a rather long training phase [31]. It's challenging to apply this method on mobile devices, especially if there is no hardware keyboard but a soft input method, like touchscreens. The hold time is a less meaningful factor and thus less applicable on mobile devices than on normal sized keyboards [18].

3.4.3 Location

"People are creatures of habit" [26], so they usually visit the same places every day like an office at work. Location based authentication exploits this very routine by tracking an individual's whereabouts. This approach is predestined for implicit authentication: It would be outrageous if the mobile device asked its user to move to a certain location in order to verify her, which would indeed become a hide-and-seek like game. However, if the user moves out of her regular action space, the phone might take measures to ensure that it's in the hands of the rightful owner. Evidently, it's necessary to have additional authentication mechanisms, since the user might actually one day go to unfamiliar places. So there are further steps to be taken.

3.4.4 Network Usage and File Access

A fairly new mechanism was proposed in 2009 by Yazij et al. [31]. They investigated if it were possible to identify a user from examining her network activities and file access behavior. So if a certain individual visits a particular website about 6 times per day, and another website about 10 times, one can build a user model for normal behavior. This approach is especially strong if an attacker does not know that there are certain patterns to be followed, in order to stay logged in. However, if the impostor knows what he has to do to mimic the victim, the system fails very soon. Yazij et al. therefore propose to use this kind of authentication in combination with other features.

3.4.5 Other Behavioral Cues

Some people do have specific **call patterns** - they call the same person every other day and are called by only a limited number of contacts on their phone. These call patterns are user specific and can also be exploited to calculate a precise authentication score [17],[26].

To conclude, an overview of the different authentication features is shown in *table 1*.

4 Functionality of Implicit Authentication

In this section, the workings and approaches of implicit authentication is explained. Although there is a separate section dealing with mobile device specific advantages and problems, the paper tries to maintain the focus on mobile devices.

4.1 Implicit vs. Explicit Authentication

Traditionally, a user authenticates herself after the device asks her for a certain proof of identity (see *section 3*). This process is called *explicit authentication*. This type of verification becomes a time-consuming procedure if it is necessary to explicitly authenticate for many different services or accounts. As we have seen in *section 3.1*, heavy users have an average of 21 passwords to remember. Typing in 21 passwords and the according user names appears to be a lot of effort and suffers from major usability issues. It seems comprehensible that users work around this problem by re-using passwords or choosing words they can easily remember. The problem is aggravated if the user is not challenged to enter a password regularly. In that case the session is kept alive for a rather long period, sometimes until the device is turned off or even longer. The Single-Sign-On (**SSO**) paradigm is supported by password managers that reduce the problem of frequent authentication [26]. Password managers therefore increase usability but drastically reduce privacy.

On the one hand, implicit authentication is designed to minimize memory challenges and time consuming verification procedures. On the other hand, continuously verifying the holder of a device brings about an increase in security, such that personal data is not compromised through session hijacking. Thus, implicit authentication might one day be able to replace explicit authentication, but the current technologies do not meet the requirements to do so. As a consequence, current research focuses on enhancing explicit authentication and thus both usability and security by adding implicitly gathered authentication cues.

Finally there are certain scenarios in which implicit authentication acts as a fraud indicator not requiring any user interaction on the actual device [26]. This can be the case when a credit card has been stolen and the thief wants to make an on-line purchase with it. An implicit authentication system could detect that the legitimate owner of the credit card is currently busy. For instance because her or she is making a phone a call in a certain location. Knowing this, the system can then take measures to prevent the purchase with the stolen credit card, because it is not plausible.

This paper mainly focuses on implicit authentication as a usability advantage and as second factor security enhancement. Fraud indication as stated above is not primarily in the scope of this work.

Feature	Capturing Method	Implicit / Explicit	Spoofing Threats	Applicability on Mobile Devices	Problems
Passcode	Hard / Soft keyboard Input	Explicit	Keyloggers, Shoulder Surfing	No constraints	Guessable passwords are still the most used
Token	Hardware Device shows a password, that expires after a short time	Mainly explicit, but implicit authentication through Bluetooth possible	None	Mobile devices are rather small, carrying around two devices seems too much to ask from a user.	Easily stolen or lost
Face & Iris	Camera	Both	Photographs of the legitimate user	Front Camera necessary (for face recognition)	Lighting situation and make-up
Fingerprint	Fingerprint reader	Currently only explicit, implicit authentication difficult	Play-Doh casts or Scotch Tape	No constraints	Injuries on the finger alter the minutiae pattern
Speech	Microphone	Both	Recordings of the user's voice	No constraints	Sickness, natural voice changes, background noise
Gait	Camera or Accelerometer	Both	Gait imitation (difficult)	No constraints	Injuries, carrying load, footwear, ground surface, being seated
Keystroke	Hard/soft keyboard	Primarily implicit, explicit possible	Typing imitation (difficult)	Possible, but difficult	Long training phase, reliability
Location	GPS or infrastructure calculated position	Primarily implicit	Informed strangers	No constraints	Traveling outside the regular scope, precision
Network/File Access	Software protocol (exemplary tool: WireShark)	Implicit	Informed strangers	File access is restricted, since the file paradigm is not too widespread on smart phones. No constraints for network usage	Precision

Table 1. Comparison of different authentication methods

4.2 The Imprinting Paradigm

In order to better understand the idea of implicit authentication, some research papers suggest to regard the relationship of human and computer as a *parent-child-relationship* found in the animal world. For example: " Geese [...] imprint on the first suitable moving object they see shortly after hatching, and will ever after treat it as their parent"[14].

Applying this imprinting paradigm to computers would allow for a more secure handling of the device. The first time the computer or mobile device is used, it imprints on the user relying on as many cues as possible to recognize her. For security reasons, not even the user should be granted access to the storage of the data model that represents her [14]. The hereby established trust relationship is stronger than using a challenge/response authentication scheme, as the user can be assured that the computer will not give away personal data freely.

Imprinting is a process that presumably cannot take place in a single moment. Rather, it is necessary to run through a special training phase, where the imprinting happens. The duration of the training phase is a critical point. On the one hand the system should work as securely as possible, which a longer training phase could ensure. On the other hand the timespan in which the device is still unprotected, respectively *less* protected, should be as small as possible. These two aspects have to be considered when taking a decision concerning the training duration. In the literature one finds suggestions that training should at least take two months to yield a False Rejection Rate (**FRR**) of 2% or less [31], which is an acceptable value in terms of security. Implicit authentication mechanisms are not active during that enrollment.

Furthermore, there exist two types of training phases, analogue to authentication: explicit or implicit training. A user-initiated training phase is regarded problematic, since users are not motivated enough to secure their devices [29]. User studies have shown that explicit enrollment of authentication data, e.g. keystroke dynamics, is a bothering task [7]. One can conclude that people would then refuse to train their devices properly, which would in turn render implicit authentication useless. So it is suggested to perform an implicit training phase, as well. This means that the device does not ask the user to take certain steps, but rather informs her, that data will be collected for a certain time. After that duration, the system becomes active [7]. If calculation and storage resources are sufficient, there is no reason to stop collecting enrollment data after this point, so that the system is strengthened.

If a person other than the legitimate owner uses the device during the training phase, the enrollment data is biased. At this point, it appears no reasonable solution to this problem has been found because research papers in this area do not treat this case at all.

4.3 Algorithms

This section shortly presents some algorithms that are used to perform anomaly detection afterwards. When going through literature, only few papers explicitly describe the functionality of the used algorithm in detail, as some of these are expected to be well known. Rather, concepts are shown, leaving room for further ideas.

4.3.1 Prerequisite: Evaluation and Metrics

In many research papers that performed user studies (with prototypes), the tested algorithms were measured and compared through False Acceptance Rates (**FAR**), False Rejection Rates (**FRR**) and in some cases through Equal Error Rate (**EER**). FAR refers to the number of times an impostor is mistaken for the legitimate user, while FRR tells how often a legitimate user has not been recognized. Both measures are expressed through a percentage relative to a total number of impostor or genuine authentication attempts. Thus, FAR and FRR are calculated as follows [13]:

$$FAR = \frac{N_{accepted\_impostors}}{N_{total\_impostors}}; FRR = \frac{N_{rejected\_genuines}}{N_{total\_genuines}}$$

In general one can state that "FAR relates to the security of the system, while FRR to the usability" [13]. Balancing those two rates while still

minimizing both is the primary goal of every authentication technique. That balance is expressed through the EER, that represents the state where FAR equals FRR [13].

4.3.2 Anomaly Detection

Given a user model and certain features (see *section 4.3.3*), the core functionality of implicit authentication are classification- and scoring algorithms, which are a sub-area of data mining. They either compare a sample to the user model or calculate a probability score.

Neural networks. When the result should be a classification (legitimate user / impostor) the use of neural network algorithms is suggested, because the yielded FARs and FRRs are usually quite low, thus more accurate [7]. They take an input vector containing the measured feature data and try to find known patterns in the user model, resulting in classified data. On the downside, the complexity of neural network algorithms is rather high especially when applied for large data sets, so they require higher CPU performance. As a consequence, this sort of classification technique is less applicable on mobile devices themselves, but potentially suitable in a powerful cloud.

Statistical and Heuristic Methods. More often, statistical algorithms are used for anomaly detection. Their major advantage are low processing requirements while maintaining acceptable results [31],[7]. They perform very fast and energy efficient even for large data sets. Exemplary algorithms in this area are K-Means Cluster, FFT, correlation and Bayesian networks - each one has its advantages in different areas. For example, gait recognition is especially suitable for correlation (see *figure 3*).

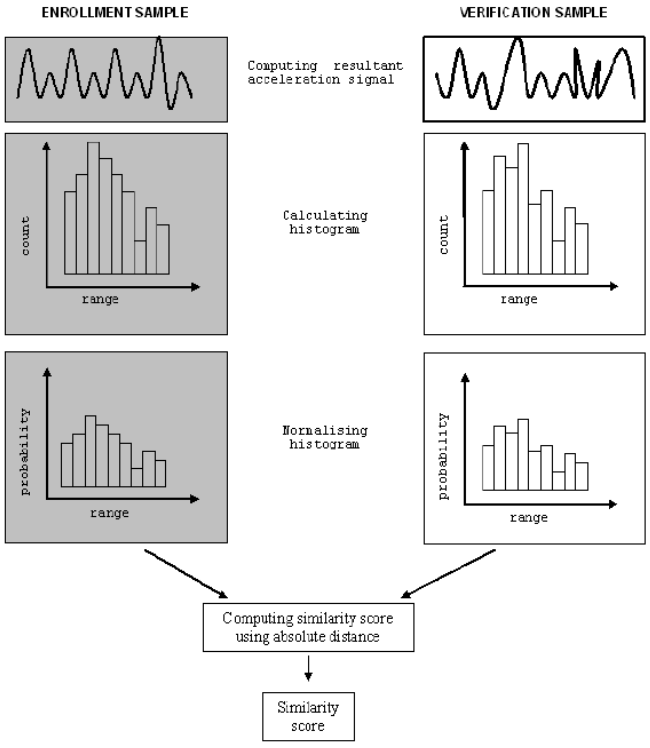

Fig. 3. The process of applying the histogram similarity method [13]

4.3.3 User Models and Scores

In order to make an authentication decision, most approaches utilize a **scoring function**. This function usually requires a previously generated **user model**, which represents the rightful user of the device. When combining multiple features (which is recommended, see *section 5.1.2*), each one of these features has to be rated individually by the scoring algorithm. Since some biometrics are more reliable than others, some algorithms weight the resulting scores, based on the feasibility and precision of the measured cues. After that, a final score is generated that is then used to either reject or accept the authentication.

4.3.4 Exemplary Scoring Functions

Jakobsson et al. [17] as well as Shi et al. [26] propose the introduction of "good" and "bad" events that influence the current authentication score. In their approaches, the user's recent behavior is described by a tuple $(t, v_1, v_2, ..., v_k)$ where t is the current time and $v_1, ..., v_k$ denote the values of independent feature-variables at time t. After computing a separate score for each feature, a function f is then used to combine these scores into a final score. Additionally, a probability density distribution is included into the scoring, meaning the score has to decrease over time. "Good calls" are made to or received from known numbers, e.g. from the contact list stored on the device. Contrarily, "bad calls" are all the others, i.e. made to unknown numbers.

Let $V_1 = time\ elapsed\ since\ last\ good\ call$, then one has to adapt the probability according to the daytime, for example. As a consequence, for someone who usually makes phone calls in the afternoon, but rarely in the morning, the score is expected to **decrease faster** in the afternoon, because phone calls during this time are more probable. This allows for a faster anomaly detection.

4.3.5 Exemplary Algorithm

If one wants to summarize how user models are built and at what point, the score calculations take place, it is helpful to illustrate the procedure with a pseudo-code algorithm. An example of such an algorithm based on work by Yazij et al. is presented in *algorithm 1* [31]:

Algorithm 1 General User Model Building and Implicit Authentication Algorithm according to [31]

```
For each user U_n do the following:
while TRUE do
    Receive data from capturing system
    if NewUser = TRUE then
        if time < TrainingDuration then
            Add log data to the data source
        else
            Build the user's profile
            NewUser = FALSE
            Send user's profile to the cloud
        end if
    else
        {user model already exists}
        Perform anomaly detection
        if abnormal behaviour then
            Take action, like re-authentication through PIN, lock device, or warn the user
        else
            continue
        end if
    end if
end while
```

5 Mobile-Device-Specific Implicit Authentication

We have seen the advantages and workings of implicit authentication in the previous chapter. These findings are now substantiated in the following sections.

5.1 Smartphones

This section is about further unique conditions of implicit authentication for mobile devices and for smart phones in particular.

5.1.1 Suitable features

As one can see in *table 1* certain features/cues are more suitable for implicit authentication on mobile devices than others. This is because mobile devices usually have less capabilities of capturing behavior through sensors and software. However, the literature proposes

a few cues that have already been evaluated or will be tried out in the future. The three major approaches so far were **gait recognition**([13],[27],[19]), **keystroke analysis**([7],[21]) and **behavior patterns**([26],[17],[9]).

First approaches relying on (mobile) gait recognition made use of an external accelerometer attached to a specific body part, e.g. to the lower leg [27]. They are now replaced by an internal gyroscope which many current smart phones possess. Keystroke recognition has mainly been evaluated on phones with hardware keyboards - touch-screen based phones are not entirely excluded, because they offer other usable characteristics: the size of the area produced by finger pressure on the screen gives hints on who is using the phone [25]. The mostly suggested behavioral patterns were user actions (e.g. phone calls or web-sites) in combination with location information. Location based continuous authentication is predestined for mobile devices.

5.1.2 Combining Multiple Features

Since single cues have rather large error rates (cf. *section 4.3.1*) it is reasonable to fuse many features to calculate an authentication score. For example, if a device collected information from 20 different cues, with independent error rates of 20%, a 2/3 vote has only a theoretical chance of 1:500,000 of taking a false decision [14]. Similar results have been presented by Yazij et al.: Their network, respectiveley filesystem, based approach had maximal FARs of 65%, respectively 94% - which are unacceptable values [31].

It is also important to have fall-back options when one feature is currently not available at all.

5.1.3 Further Considerations

At this point, there is no "real" solution or software for implicit authentication on mobile devices, that one can buy and install on one's device, which stands in contrast to explicit authentication that has a lot of different solutions to offer. However, a stable implementation might not be too far ahead, since the algorithms and techniques already exist. Several prototypes have been evaluated in exemplary studies [9].

One challenge that has to be taken are hardware constraints: More devices would profit from implicit authentication if the functionalities are realized without having to add hardware to the handset. The suggested Trusted Platform Modules are not available in every device - so this approach seems less promising, although the security would increase.

5.2 Local vs. Remote Calculation

Basically, the scoring calculation can take place on the device itself or on a more powerful remote system. Due to the limited calculating capacity on mobile devices (see *section 7*), it is recommended by the majority of recent research papers to make use of *Remote Attestation* [21]. According to this approach, log data and variable states are repeatedly packed and then transmitted to a remote server. Naturally a nearly permanent mobile Internet connection is required, because data is transmitted in short time intervals. Since this server possess higher capabilities, the calculation of the authentication score is performed quickly. The result containing the decision if the session ought to be quit or not is immediately sent to the phone, which can take the according measures.

It has to be noted, that all data transmitted through the network is highly sensitive. Thus, it is mandatory to establish trusted connections through encrypted channels [21]. Furthermore, suggested solutions also demand the above mentioned *Trusted Platform Module* (*TPM*) in order to ensure that the endpoints of the communications have not been compromised and the data is valid.

5.3 Cloud Based Implicit Authentication for Mobile Devices

A recent development enhancing the principle of remote attestation are cloud based authentication frameworks. One of the reasons for the increased demand of cloud computing could be the ongoing sales of smart phones. These devices offer access to information that is spread across the Internet. It is necessary to store data externally because typical handset storage capacity is rather small compared to desktop PCs. In order to enable complex calculations and applications, mobile devices outsource difficult tasks. Consider Anti-virus software for smart phones: the analysis of a file can be very intensive and is likely to use up a lot of battery, as well as CPU capacity. However, it would be preferred if this was accomplished through a background process, which ought not to slow down the phone. In this scenario an external service, which often is realized using cloud technology, receives the signature of a file, that has to be checked. As soon as all the necessary data has been transmitted, the cloud service performs the analysis-job and sends back the result to the device, e.g. "the file is malicious" or "no suspicious signatures found". Although this procedure could be implemented deploying a single server, cloud computing offers functionalities and opportunities, that traditional client-server-architectures lack: High dynamic scalability, data transparency and high throughput computing armbrust2009above. This comes in handy, when there are many thousands of users accessing the authentication framework services from the cloud.

Such a framework was proposed by Chow et al. [9]. Their so-called *TrustCube* uses **policies** to support authentication decisions, which are taken based on calling patterns, SMS activity, website access and location. Services are modularized into a star-shaped topology generating privacy benefits, as only the central node needs to collect user-specific data. Among these services one finds data aggregators (i.e. the implicit authentication server), an authentication engine and several authentication consumers. The client software was built for the Android operating system. The elementary authentication procedure is as follows: After the handset has collected **user data** for a specific time window, the data is packed and reported to the data aggregator (and afterwards deleted from the device, to free memory). After that, the client agent collects **phone-specific information**, such as firmware version and running applications. Finally the authentication service calculates a score and makes a decision based on the give **policy**. Such a policy may look like this: "(1) the device should run Android 2.1 update 1 or above AND the WiFi SSID should be "hospital" AND only default and hospital applications can be installed; (2) the minimum score to view medical data is 800; (3) if the authentication score is below the minimum, the user must use a PIN pad to further authenticate."

5.4 Implicit Authentication on Other Portable Devices

Some solutions presented in the literature do not primarily target smart phones. Yazij et al. developed a prototypical implicit authentication framework that runs on laptop computers [31]. These devices offer a mature file-system implementation, which the researches used for authentication beside network usage and location. In their conclusion they stated that in the future they as well will focus on smart phones.

Lastly, Shi et al. [26] and Jakobsson et al. [17] suggest the use of implicit authentication for portable medical devices. They think of a digital clipboard that doctors carry around. These devices contain highly sensitive information, which on the one hand has to be accessible very fast in case of an emergency - typing in a password might waste critical seconds and continuous authentication could speed up getting to the information. On the other hand has to be protected from unauthorized access, for example the data for a certain patient should only be available for doctors of a specific position at a hospital.

6 Adversarial Models

Any form of authentication tries to defend a certain device, account, area etc. against foreign attacks to protect sensitive data. The attackers - or adversaries - possess different incentives and capabilities, which have to be specified in order to be able to shield personal data. For implicit authentication, some particular characteristics of possible adversaries have to be taken into account. Shi et al. provide a detailed adversary model in which they characterize adversaries by roles, incentives and capabilites [26]. Their model is summarized below.

6.1 Roles

If a smart phone, for example, gets into the hands of someone other than the user, this person is not necessarily an *attacker*. In certain cases, *Friends or co-workers* got hold of the handset. When at home, *family members* can easily access phones not originally belonging to themselves. *Strangers* might have stolen the device or found it in a public place where the user accidentally left it. Finally, *enemies or competitors* could try to reap information for political or industrial espionage. This leads right to the different types of incentives, that an adversary might possess.

6.2 Incentives

When it comes to the question "Why would someone want to get hold of my phone?", one can describe certain incentives that motivate adversaries to try and capture a device. If an adversary wants to gain *financial advantages* this may happen for the following reasons: (1) He can make free phone calls or use the Internet at the owner's expense. Or he uses the device for entertainment, maybe because he does not own one himself. (2) He might sell the device or parts of it. (3) If personal data, such as emails, reveals access information for bank accounts, the attacker is inclined to take advantage of this. Thus, he may perform on-line purchases or transfer money to his own account at the cost of the owner of the stolen device. The latter is a major issue and the most difficult to resolve.

Another incentive is sheer *curiosity*. Family members, co-workers or the significant other might want to know read a user's emails or SMS messages. Possibly the browsing or phone call history are also interesting for others. The issue becomes more serious if an adversary is driven by espionage. He might want to obtain sensitive data concerning business matters.

Additionally, some illegitimate users are not willing to take any advantage, but to *sabotage* a victim, either financially or in reputation matters. This means that they buy things on-line just to cause trouble for their victim. Aside from that, the phone can be used to publish embarrassing remarks or pictures on social networks - recently informally termed "frape", a combination of the words 'Facebook' and 'rape' [23].

Finally, a stolen phone can cater for the thief's *anonymity*. He could access illegal web-sites or share illegal material and not get caught, because the traces lead to someone else.

6.3 Capabilities

Capabilities are established especially for implicit authentication. While every attacker being asked for a password immediately knows, that there probably exists one, this is not the case if the user is verified implicitly. However one can attribute three different capabilities to an adversary.

The *uninformed stranger* is not aware, that implicit authentication takes place. Thus, after capturing the device, the person is likely to use the handset as his own, or use it to achieve different incentives as described above.

In certain cases, implicit authentication becomes more challenging when dealing with an *informed adversary*, who is aware of the existence of implicit authentication. It might be easier to imitate certain features, if the attacker knows the legitimate owner in person. The more features are used for verification, the more difficult it becomes even for the informed adversary to game the system.

Lastly, if an attacker infects the handset with *malware*, behavioral patterns can be logged and mimicked afterwards, which would invalidate the whole implicit authentication system. There are certain measures to be taken to protect the implicit authentication software from malware, like installing Anti-Virus solutions.

7 Limitations of Implicit Authentication on Mobile Devices

This section covers the deployability of implicit authentication on mobile devices. It remains open what factors do have to be considered when designing and programming continuous authentication software that is targeted at mobile devices.

7.1 Battery Usage

Implicit authentication happens in the background, therefore implementations use background processes to provide continuous authentication. However, these processes use up a part of the battery capacity, so the phone has to be recharged more often. This in turn reduces the battery lifetime, since charging cycles are not infinitely repeatable.

A detailed calculation of energy consumption through implicit authentication was carried out by Yazij et al. [31]. They examined how implicit authentication relying on network- and file access plus location influenced the battery lifetime of a laptop. It was shown, that remote attestation consumes less energy: Only **6.6%** were used up by implicit authentication mechanisms if deploying an external service whereas 42.6% of battery capacity were wasted if the anomaly detection took place on the device itself.

7.2 Calculation Speed / Detection Latency

Another problem with implicit authentication is detection latency. The paradigm states that authentication happens unobtrusively, so that a secured device should not interrupt the user and divert her from tasks. Since other factors have impact on how often the authentication score can be calculated, there still is a certain time window, where the handset can be used by an attacker without the system noticing. If battery lifetime and processor performance increase one can minimize the detection latency by locally calculating authentication scores.

7.3 Data and Traffic Amount

Collecting data eventually leads to storing it somewhere on the device - either only temporarily or permanently. The log data becomes larger if multiple cues are used for authentication, which we have seen, is recommended. In a user study conducted by Yazij et al. participants generated log data between 80MB and 25GB within two weeks on laptop computers. However, probably none of current handsets could cope with log data above 10GB. Besides, users actually would like to use the limited space for features for which perceived usefulness is a lot higher - log data has no obvious benefits.

One already addressed solution are data aggregators or external storage in general. This in turn suffers from the fact that the data has to be transfered from the mobile device to the cloud, for example. Mobile phone contracts usually cover a certain traffic amount that is free of extra charge. It is probable that transferring a lot of log data through UMTS or LTE networks might eventually lead to higher monthly costs, which some users will not be willing to take. Thus, a good balance between locally stored data and traffic has to be found. In order to minimize traffic, one could also find better ways to compress the data. If the network isn't available at all, for example because the user travels to a foreign country and refuses to pay extra fees for roaming, the calculation of the authentication scores will have to take place on the device. Otherwise implicit authentication is deactivated.

7.4 Account Sharing

In certain cases, a user might allow a friend to use her phone, for example to look up something on the Internet or for entertainment. If the phone uses implicit authentication mechanisms, the phone is able to detect the illegitimate user and possibly shuts down. This process might even reoccur when the correct PIN is entered afterwards by a different person. Therefore solutions to this problem have to be found, without burdening the user of switching on and off the authentication services.

7.5 Reliability

Even though combining multiple features to authenticate a user implicitly yields relatively good results, there still remains a certain percentage of impostors who are mistaken for the legitimate user. In other words, an FAR of 0% is desirable but in most cases not possible. The same holds true for usability matters, where an FRR of 0% would mean no bothering re-authentication, but one cannot avoid it - for the moment.

8 Conclusion

In this paper, a detailed view of authentication and implicit authentication has been described. The most valuable features that implicit authentication relies on are gait, keystroke and behavioral patterns such as phone call- or web-surfing-activity. Although the computational power of smart phones rapidly increases, outsourcing implicit authentication tasks, like decision taking processes, seems the most reasonable at the moment. Specially designed frameworks for cloud computing might become a de facto standard in the near future, unless users refuse the new technology. Fortunately, current studies indicate that users would welcome the establishment of implicit authentication [7], [12]. Also, creating software that runs on as many operating systems as possible is a major task that the developers are yet to face.

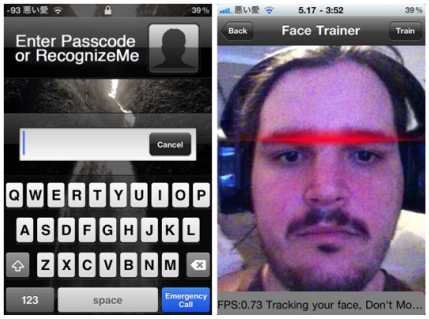

Fig. 4. The RecognizeMe iPhone app uses the front camera to explicitly authenticate a user [2]

Although implicit authentication might not be able to replace explicit authentication entirely, and although parts of it show some disadvantages, the principle is really promising in terms of both security and usability.

The author would like to suggest face recognition techniques for implicit authentication on mobile devices, which have not been under research recently. As new generation smart phones have additional cameras oriented to the user's face, one could think of taking pictures every few minutes to see whether the face matches the legitimate user. There is already an iPhone app called RecognizeMe (see *figure 4*) that can authenticate the user through the iPhone 4's front camera, given it is jailbroken [2]. It seems rather easy to enhance this application to become suitable for implicit authentication.

References

[1] H. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S. Makela. Identifying people from gait pattern with accelerometers. In *Proceedings of SPIE*, volume 5779, page 7, 2005.

[2] Apocalipse. RecognizeMe iPhone App on Cydia. `http://modmyi.com/cydia/com.apocolipse.recognizeme`. visited 18.06.2011.

[3] Apple. Set a Passcode Lock with Find My iPhone. `http://www.apple.com/mobileme/news/2009/09/set-a-passcode-lock-with-find-my-iphone.html`, 2009. visited 02.06.2011.

[4] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[5] J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Combining biometric evidence for person authentication. *Advanced Studies in Biometrics*, pages 1–18, 2005.

[6] R. Brunelli and D. Falavigna. Person identification using multiple cues. *IEEE transactions on pattern analysis and machine intelligence*, 17(10):955–966, 1995.

[7] A. Buchoux and N. Clarke. Deployment of keystroke analysis on a smartphone. In *Proceedings of the 6th Australian information security management conference. Perth, Western Australia: SECAU-Security Research Centre*, pages 40–47, 2008.

[8] S. Card, T. Moran, and A. Newell. Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology*, 12(1):32–74, 1980.

[9] R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shi, and Z. Song. Authentication in the clouds: a framework and its application to mobile users. In *Proceedings of the 2010 ACM workshop on cloud computing security workshop*, pages 1–6. ACM, 2010.

[10] J. Cutting and L. Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the psychonomic society*, 9(5):353–356, 1977.

[11] I. Damousis, D. Tzovaras, and E. Bekiaris. Unobtrusive multimodal biometric authentication: The humabio project concept. *EURASIP journal on advances in signal processing*, 2008:1–11, 2008.

[12] S. Furnell, N. Clarke, and S. Karatzouni. Beyond the pin: Enhancing user authentication for mobile devices. *Computer fraud & security*, 2008(8):12–17, 2008.

[13] D. Gafurov, K. Helkala, and T. Søndrol. Biometric gait authentication using accelerometer sensor. *Journal of computers*, 1(7):51–59, 2006.

[14] R. Greenstadt and J. Beal. Cognitive security for personal devices. In *Proceedings of the 1st ACM workshop on AISec*, pages 27–30. ACM, 2008.

[15] G. Hayday. Security nightmare: How do you maintain 21 different passwords. `http://tinyurl.com/silicon-security-nightmare`. visited 03.07.2011.

[16] A. Jain, F. Griess, and S. Connell. On-line signature verification. *Pattern recognition*, 35(12):2963–2972, 2002.

[17] 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*, pages 9–9. USENIX Association, 2009.

[18] S. Karatzouni and N. Clarke. Keystroke analysis for thumb-based keyboards on mobile devices. *New approaches for security, privacy and trust in complex environments*, pages 253–263, 2007.

[19] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. Makela, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *IEEE international conference on Acoustics, speech, and signal processing, 2005. Proceedings (ICASSP'05).*, volume 2, pages ii–973. IEEE, 2005.

[20] Motorola. Fingerprint reader of the motorola atrix 4g. `http://mediacenter.motorola.com/ImageLibrary/Detail.aspx?MediaDetailsID=1472`. visited 09.06.2011.

[21] M. Nauman and T. Ali. Token: Trustable keystroke-based authentication for web-based applications on smartphones. *Information security and assurance*, pages 286–297, 2010.

[22] PayPal. `http://www.paypal.com`. visited 03.07.2011.

[23] Phailure. Definition of frape on urban dictionary. `http://www.urbandictionary.com/define.php?term=Frape&defid=2463827`. visited 17.06.2011.

[24] RSA. SecurID. `http://www.rsa.com/node.aspx?id=1156`. visited 03.07.2011.

[25] H. Saevanee and P. Bhattarakosol. Authenticating user using keystroke dynamics and finger pressure. In *Consumer communications and networking conference. CCNC 2009. 6th IEEE*, pages 1–2. IEEE, 2009.

[26] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. *Information Security*, pages 99–113, 2011.

[27] M. Tamviruzzaman, S. Ahamed, C. Hasan, and C. O'brien. epet: when cellular phone learns to recognize its owner. In *Proceedings of the 2nd ACM workshop on assurable and usable security configuration*, pages 13–18. ACM, 2009.

[28] S. Tulyakov, F. Farooq, P. Mansukhani, and V. Govindaraju. Symmetric hash functions for secure fingerprint biometric systems. *Pattern Recognition Letters*, 28(16):2427–2436, 2007.

[29] A. Whitten and J. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium*, pages 169–184. Citeseer, 1999.

[30] S. Wiedenbeck, J. Waters, L. Sobrado, and J. Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the working conference on advanced visual interfaces*, pages 177–184. ACM, 2006.

[31] S. Yazji, X. Chen, R. Dick, and P. Scheuermann. Implicit user re-authentication for mobile devices? In *Ubiquitous intelligence and computing: 6th international conference, Brisbane, Australia, July 7-9, 2009, Proceedings*, page 325. Springer-Verlag New York Inc, 2009.