Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

Magnus Rolfsøn

# An evaluation of authentication methods for solutions that require a high degree of both security and user-friendliness on mobile phones

Master's thesis in Information Security
Supervisor: Patrick Bours, Jørn Magne Raastad, Mads Egil Henriksveen
June 2019

NTNU
Norwegian University of Science and Technology

buypass

Magnus Rolfsøn

# An evaluation of authentication methods for solutions that require a high degree of both security and user-friendliness on mobile phones

Master's thesis in Information Security
Supervisor: Patrick Bours, Jørn Magne Raastad, Mads Egil Henriksveen
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

# Preface

This thesis concludes my master's degree in the field of Information Security at the Norwegian University of Science and Technology (NTNU) in Gjøvik. The thesis was performed during the spring semester of 2019 and aims to find the best authentication method for use with mobile phones when considering both security and user-friendliness. The general subject of the thesis was given by Buypass, while the research questions was formulated together with my supervisor at NTNU, Patrick Bours. The thesis was written in cooperation with Buypass, who provided two supervisors of their own, helping and guiding me through the thesis. Performing the research was difficult, but in the end we were able to answer the research questions we defined.

01-06-2019

# Acknowledgment

I would like to thank Patrick Bours for being my supervisor, and for all the help and quick replies he provided during the thesis.

I would like to thank Jørn Magne Raastad and Mads Egil Henriksveen for being my supervisors at Buypass, and for all the help they have provided ranging from meetings to letting me borrow newly bought phones so that I could test presentation attacks on them. Without Buypass, I would also not have this thesis, so a big thanks to them for providing the idea behind the thesis.

I would like to thank my friends and family for all the support. A special thanks to Kristin Schnell Rolfsøn and Ragnhild Søhol for all the help and support you have given.

Finally, to all of you, one last thanks, I do not think I could have finished this thesis without your help, and for that I am grateful.

M.R.

# Abstract

With the many different authentication methods that are available on mobile phones, finding the best one for can be challenging. Some situations calls for security over user-friendliness, while in other situations, having the most user-friendly authentication method might be the best. This thesis aims to find the best authentication method when considering both security and user-friendliness.

The security of the different authentication methods included in this thesis was found by searching literature, though there were problems finding literature and information for some phone manufacturers about their biometric accuracy and performance. A questionnaire was distributed to collect information about the length of PINs and passwords of users, as well as how user-friendly users think different authentication methods are. For some authentication methods, the amount of answers were satisfying, but for other less popular authentication methods the amount of answers were too low to really draw any meaningful data that could also be applied to a larger population. The best authentication method when considering both security and user-friendliness was found to be 3D facial recognition, with it having a very low false accept rate and high user-friendliness score.

Three presentation attacks were also conducted during the thesis, two on fingerprint recognition and one on 2D facial recognition. Attacks on fingerprint were mostly unsuccessful, while the attack on 2D facial recognition were mostly successful.

# Sammendrag

Med de mange forskjellige autentiseringsmetodene som er tilgjengelig på mobiltelefoner, kan det å finne den beste være en utfordring. Noen situasjoner kan kreve sikkerhet over brukervennlighet, mens i andre situasjoner kan det å ha den mest brukervennlige autentiseringsmetoden være det beste. Denne avhandlingen har som mål å finne den beste autentiseringsmetoden med tanke på både sikkerhet og brukervennlighet.

Sikkerhetsgraden til de forskjellige autentiseringsmetodene inkludert i denne avhandlingen ble funnet ved hjelp av litteratursøk, selv om det oppstod problemer med å finne litteratur og informasjon om nøyaktigheten og ytelsen til noen mobilprodusenters biometriske autentiseringsmetoder. I sammenheng med avhandlingen ble det laget en spørreundersøkelse hvor respondenter ble spurt om hvilke autentiseringsmetoder de bruker, hvor lange PINs og passord de har, samt hvor brukervennlig de føler at autentiseringsmetodene de bruker er. For noen av autentiseringsmetodene var antall svar tilfredsstillende, men for de mindre populære autentiseringsmetodene var det for få svar til å virkelig få noe god data som også kunne blitt brukt for større populasjoner. Den beste autentiseringsmetoden med tanke på både sikkerhet og brukervennlighet ble vist å være 3D ansiktsgjenkjenning, da denne autentiseringsmetoden har både lav "false accept rate" og scoret høyt på brukervennlighet.

Tre presentasjonsangrep ble også utført under avhandlingen, hvor to av disse var mot fingeravtrykksgjenkjenning og et på 2D ansiktsgjenkjenning. Angrepene på fingeravtrykk var hovedsakelig lite vellykket, og angrepet på 2D ansiktsgjenkjenning var hovedsakelig vellykket.

# Contents

# List of Figures

# List of Tables

# Listings

# 1 Introduction

## 1.1 Topic covered by the thesis

Within this thesis, we seek to make a ranked list of the best authentication methods to be used in mobile phones that need a high degree of security and user-friendliness. To do this, we take a look at different authentication methods that can be used with mobile devices, their performance, attacks and countermeasures on these systems, as well as factoring in user-friendliness.

## 1.2 Keywords

Biometric, biometric security, authentication methods, PIN, password, pattern authentication, fingerprint recognition, facial recognition, iris recognition, presentation attacks, presentation attack detection, security evaluation, user-friendliness

## 1.3 Problem description

Secure authentication in digital systems is a rapidly developing area. Already known biometric modalities such as fingerprint recognition and facial recognition have made authentication for services that require it much easier for the end user, for example fingerprint or facial recognition on smartphones. The task will be to evaluate the security level and user experience of different available authentication methods and compare them. The thesis will focus on solutions for mobile telephones.

## 1.4 Justification, motivation and benefits

As security is more important now than ever, the use of secure and user-friendly biometric authentication methods, instead of, or complimentary to passwords, is beneficial to everyone. Weak PINs and passwords are prevalent, and seeing that a lot of applications restrict a user to a 4-digit PIN, which is not secure without any other form of security measure, like a limited number of tries before being locked out. Some biometric authentication methods are also better than others, and more user-friendly. By ranking the different authentication methods by their security and user-friendliness, we can help people choose the right authentication method for their system.

## 1.5 Research questions

In this section we will discuss the research questions for this project and also give a number of sub questions that help us answer our main research question.

**What is the best authentication method on a mobile device when considering both security and user-friendliness?**

*We want a system that combines both security and user-friendliness and not compromises one for the other. Without user-friendliness, the adaptation of the authentication system might not be as high, and therefore degrade the security of the application. With an authentication system with high user-friendliness, but low security, the whole security of the system is at stake, because getting into someone else's system without being authorized will be easier. So, the best authentication system will have to have some form of balance between the two.*

Based on this research question we have derived the following sub questions:

**How can we measure the security of different solutions?**

*We have to consider for example if there are any weaknesses of the specific system, if there are any known attacks against a specific system, and how the data is stored on the device. How easy or hard it is to perform the various attacks on the system will also play a role.*

**When considering user-friendliness, how is the ease of use, is the system intuitive, fast and reliable?**

*If the authentication system is hard to use, users will likely not adapt that authentication method, and might settle for an authentication method with a lower degree of security. The same holds for how intuitive, fast and reliable the authentication system is.*

## 1.6 Planned contributions

In this project we will be compiling a list of the best authentication methods for use in solutions that need a high degree both security and user-friendliness. This list can be used to select the best authentication system based on the level of security and user-friendliness that is wanted. A questionnaire about user-friendliness will also be done during the project.

## 1.7 Choice of methods

In this section we are going to describe the methods that will be used during the project. The first task will be to conduct a literature study, continuing the work we started in the project planning. We will then describe the questionnaire, small scale testing, experiment and performance testing and improvement.

### 1.7.1 Literature study

The literature study will be an important part of this project, as a moderate part of the time used in this project will be spent finding literature that describes strengths and weaknesses of different authentication methods that are suited for use in mobile devices. To find this literature, different literature databases like IEEE explore, ACM digital library and also some Google Scholar will be used. To help show how some attacks is done, YouTube has also been utilized. This work will be qualitative, and can help us answer both our research sub-questions.

### 1.7.2 Questionnaire

During the project, a questionnaire will be made and distributed. This questionnaire will contain questions that will give us an answer to how user-friendly different authentication systems for mobile devices is perceived. As user-friendliness can be highly subjective, it can be interesting to see if the age or technical competence can be a factor in how people perceive the user-friendliness in a system. This questionnaire will also help us in the development of our function.

To get meaningful results, we would like at least 100 people to answer the questionnaire. Most likely, most of the participant will be using fingerprint and face recognition along with a PIN in case of too many false rejects in a row, as those are the most known biometrics characteristics when it comes to mobile devices, and that not all older mobile phone models offer anything other than fingerprint and facial recognition. If too few participants use or can answer on the user-friendliness of iris recognition or passwords, a solution can be to find people with a phone that offers this biometric, or only password protect their phone, get them to try it out and let them offer their thoughts after a period of using iris recognition or passwords to unlock their devices. This portion of the project will also be quantitative as far as it allows, but as some interviews might be needed there might be some qualitative portions.

### 1.7.3 Data analysis

After we have collected enough data from the questionnaire, we will perform some analysis on the user-friendliness of authentication methods. Some of the questions will be numerical, so that we can calculate the median and average of the scores given to us. With this data we will get an understanding of how user-friendly different authentication methods are.

### 1.7.4 Small scale testing

To add to our literature study, some small scale testing will be conducted. This testing can include doing our own presentation attacks against some of the popular authentication methods with known attacks, to see if we also can get the attacks to be successful, or try to gain access to a password protected system using a dictionary attack.

### 1.7.5 Function creation

A part of this project is to create a function that determines the best authentication method based on both security and user-friendliness. the user of the function can input their desired level of security and user-friendliness. The work with this function will help us answer our main research question, finding the best authentication method with a high degree of security and user-friendliness. The work in [11] is similar to what we want our function to, though our approach will be a little different since we are creating a function instead of a framework.

Figure 1 shows how the function will go through the authentication methods and add them to the ranked list based on the user-set security and user-friendliness criteria.

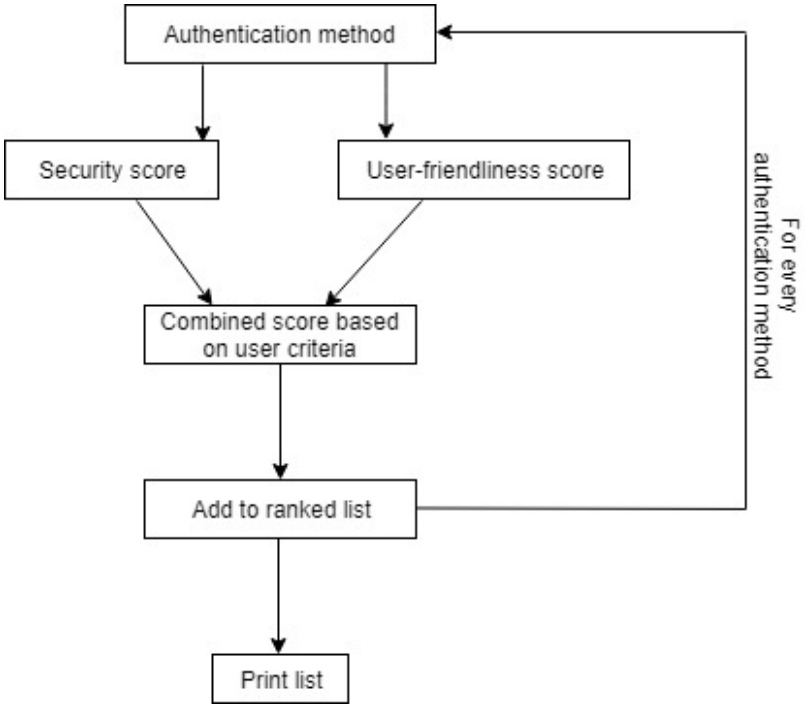


Figure 1: Diagram of the function

# 2 Background and related work

This chapter looks into the related and background work that is needed for further reading this thesis. The chapter starts with section 2.1, giving an overview over the categories different authentication methods can be put in. In section 2.2 to section 2.6 common available authentication methods are explained in further detail, as well as attacks and countermeasures for the respective authentication method. Section section 2.8 looks into research on ranking authentication methods. Very little research have been done on the subject of ranking authentication methods, so the information found in this section will be helpful to us.

## 2.1 Authentication categories

The aim of authentication is verifying a claimed identity. The common way of dividing authentication methods is into three groups.

- **Something you know - secrets**
  Passwords, PINs, pass phrases, pass images, etc.
- **Something you have – tokens**
  Keys, smartcards, USB sticks, etc.
- **Something you are – biometrics**
  Fingerprints, facial recognition, iris recognition, keystroke dynamics, etc.

### 2.1.1 Knowledge-based authentication

In a knowledge-based authentication system, the users of the system is asked to answer specific questions that have been agreed on by the user and the system beforehand. Common authentication methods from this category is passwords and PIN codes. Knowledge-based authentication methods are becoming less suitable for authentication as time goes on, both because of sophisticated social engineering attacks, and as the processing power of computers increase, the time it takes to run dictionary and brute-force attacks decrease.

For the knowledge-based category, passwords and PINs are the most common methods. The strength of passwords and PINs is often measured using information entropy. Entropy is the measure of how unpredictable a given phrase is, and is measured in bits. The pros of these kinds of authentication methods is that increasing the entropy of the phrase is easily done by making it longer, but the con is that with increasingly long passwords and PINs, the harder it is to remember. With the best-practice of having a different password for each account you have, random passwords of 10+ characters quickly stack up to be unmemorable.

The measure of entropy also only works when the passwords or PINs are totally random, which are usually not the case with user-chosen passwords and PINs [12]. If passwords are not random,

but long enough and using enough character sets, the measure of entropy can give a false sense of security, as the measure does not take into account the guessability of non-random strings and can be insufficient in visualizing a password's resistance against intelligent guessing attacks [13]. Because of this, researchers have suggested using guessing entropy as a measure of strength instead. The advantage of this metric is that it takes into account the knowledge of how users tend to build their passwords that real attackers might have, but the results of this metric depends on the chosen setup [13]. More on entropy in subsection 2.2.2.

### 2.1.2 Possession based authentication

For the possession based category, smartcards and bank cards are regular methods to use. Everyone has experience with this form of authentication one way or another, as we use our bank cards on a regular basis, though bank cards are combined with a 4-digit PIN. A lot of companies and educational institutes have started to use smartcards as access control to different rooms and at the entrance to the company. The pros of this category is that it is easy to use, and there is nothing to remember other than the token itself. The cons are that it is potentially easy to steal if you are not paying attention, and that anyone who has the token can impersonate you to the system if there is no other authentication method coupled with it.

### 2.1.3 Biometric based authentication

For the biometric based authentication, fingerprint and facial recognition is the most known biometric modalities used in mobile phones. Some phones also offer iris authentication, though the use of said authentication method is not very popular.

For a biometric authentication system to be practical and and reliable, the system should score high on 7 different characteristics. These characteristics are Universality, Uniqueness, Performance, Permanence, Collectability, Convenience, Acceptability and Security [9].

- **Universality**
  - Every person should have the characteristic. For example, there are very few people who are missing all ten fingers.
- **Uniqueness**
  - Indicates how unique the characteristic is, and is measured in False Match Rate (FMR).
- **Performance**
  - It should be possible to achieve a good recognition accuracy, speed and resources required to the application.
- **Permanence**
  - The characteristic's matching features should be stable over a period of time.
- **Collectability**
  - The characteristic should be easy to collect and measured quantitatively.

- **Acceptability**
  - The general public has to be willing to give away the characteristic for measuring and collection.
- **Circumvention**
  - Resistance to circumvention. How easily the system resists fraudulent methods after it is tested and proved.

**Biometric system errors:**
Because two samples from the same biometric characteristic like a user's right index finger is not exactly alike, biometric systems can not operate the same way as passwords do, where it is either correct or wrong, but instead uses matching scores and thresholds to quantify the similarity between the input and the stored template. Imperfect image conditions (e.g., sensor noise and dry fingers), changes in the user's physiological or behavioral characteristics (e.g., cuts or bruises on the finger), ambient conditions (e.g., temperature and humidity), and a user's interaction with the sensor (e.g., finger placement) can change the matching score outcome. A higher matching score means that the system is more certain that the input and the template comes from the same person, while a lower matching score is less certain that the two samples are from the same person. Because biometric systems operates with matching scores, a threshold for when a score is high enough to constitute a match is needed. For a biometric security system, two distributions of matching scores are made, one genuine distribution and one impostor distribution [9]. The two distributions are however overlapping, so a threshold needs to be set. The threshold constitutes how high the similarity score has to be for the system to count it as a match. Moving the threshold up will result in a lower FMR, but will in turn increase the FNMR of the system.

The FMR and FNMR of a system is related to the algorithm that the system uses. When measuring the errors of the entire system as a whole, false accept rate (FAR) and false reject rate (FRR) are used instead. These measurements take into account errors in the whole system, not only errors in the algorithm itself.

## 2.2 PIN/Password

PINs or passwords can be a great way of securing devices and user-accounts. Though passwords especially can be very secure under the right circumstances, they are very prone to "user error", where users choose PINs or passwords that are easy to guess for attackers as well as reusing their passwords across multiple accounts and devices. For a password to be as secure as possible, it should consist of random characters using at least one lower-case, upper-case, digit and special character, and be of a length of at least 8 characters. It is also recommended to use a different password for every account you have. This quickly stacks up to being unmemorable, thus many users reuse their passwords across multiple accounts. Because of all this there are multiple problems with user-chosen passwords. They are shown to generally be too short to resist attacks [14, 15, 13, 16]. It is also shown that users tend to put upper-case letters at the start of the password and append

numbers or special symbols at the end of the password [17, 18]. Users also tend to make their passwords around words or phrases, like names [19] and dates [20]. To make matters worse, when choosing passwords, some users also choose passwords based on the site they are visiting, for example putting the site name inside the password string [18]. When basing passwords around multiple words put together, users tend to choose semantically related words [21, 15], which makes the search space for attackers a lot smaller. Password reuse is also a prominent problem, with users having on average 6.5 passwords for 25 accounts [22].

To limit such flaws in user-chosen passwords, there should be a length requirement when creating a password, as well as checking the chosen password against a predefined list bad passwords and ask the user to choose another password if it is found in the list. This list can for example contain passwords obtained from earlier breaches, dictionary words, repetitive or sequential characters or context-specific words like the username, the websites name etc [23]. Another good of securing all your passwords is making use of a password manager. A password manager securely stores all your passwords in a database that can only be unlocked by a master password. Password managers also often come with a password generator that the user can use to make random passwords however long they want, and the user does not have to remember them or be part of the process of putting the password together. The user should make the master password a long, random and hard-to-crack password, but only having to remember one such password is much better than trying to remember tens to hundreds of different passwords and where they belong. In 2007, Florêncio and Herley found that on average, each user has 25 user accounts [22]. It would be likely that this number is even higher today, as more and more services are online and needs a user account. A lot of systems also put a limit on how many times a user can try to log into or their account before the account gets locked and either forces the user to wait a period of time or take extra steps to log in to the account. This is especially true for phones, where all lock screens deactivate the ability to enter another PIN or password if the PIN or password is entered wrong too many times. This deactivation time usually scales non-linearly, thus severely limiting the brute-force and most dictionary attack capabilities.

### 2.2.1 Attacks

Attacks against PIN and password security systems range from extremely simple to advanced dictionaries based on information specific to a user. The simplest attack against these systems is a brute-force attack. This attack tries every combination possible, and is essentially randomly guessing PINs or passwords. This builds on itself until the correct password is found. The brute-force attack is on average the slowest kind of attack. A smarter attack against PIN and password security systems is the dictionary attacks. These attacks are built to try the most likely PINs and passwords based on previous successful attacks and analysis. For an even stronger dictionary attack, information about the target user is collected and incorporated into the dictionary.

Over the years many databases containing sums of millions of PINs and passwords have been breached and published online. Nick Berry from Data Genetics compiled some of these databases looking for which PIN codes get chosen most often. In his database consisting of almost 3.4 million 4

digit PIN codes, 10.7% of them were the PIN code "1234". If the PIN codes were uniformly randomly distributed, the top 20 PINs would account for 0.2% of all the PIN codes available, but his findings show that the top 20 PIN codes account for 26.83% of the PINs in the database [6]. As shown in table 2.2.1 most of the top 20 PINs are patterns of some sort, like ascending, descending, repeating of numbers or keypad patterns. To have a 20% chance of guessing a PIN, only 5 guesses is needed. One third can be guessed by trying only 61 different combinations, while only 426 different PINs are needed to guess 50% of the entire dataset [6].

| | **PIN** | **Freq** |
|---|---|---|
| #1 | 1234 | 10.713% |
| #2 | 1111 | 6.016% |
| #3 | 0000 | 1.881% |
| #4 | 1212 | 1.197% |
| #5 | 7777 | 0.745% |
| #6 | 1004 | 0.616% |
| #7 | 2000 | 0.613% |
| #8 | 4444 | 0.526% |
| #9 | 2222 | 0.516% |
| #10 | 6969 | 0.512% |
| #11 | 9999 | 0.451% |
| #12 | 3333 | 0.419% |
| #13 | 5555 | 0.395% |
| #14 | 6666 | 0.391% |
| #15 | 1122 | 0.366% |
| #16 | 1313 | 0.304% |
| #17 | 8888 | 0.303% |
| #18 | 4321 | 0.293% |
| #19 | 2001 | 0.290% |
| #20 | 1010 | 0.285% |

Table 1: Frequency of the top 20 most used 4 digit PIN codes [6]

While searching for 4 digit PIN codes, Berry also found many PIN codes with other lengths. The top 20 most popular PINs for these different lengths is shown in table 2.2.1. In theory it should be much harder to guess a 5 digit PIN than a 4 digit PIN, as there are 90,000 more possibilities to choose from, but in table 2.2.1 we see that the accumulative frequency percentage of the 5 most used 5 digit PINs is 31.509% against only 20.552% for 4 digit PINs. This means that an attacker can break into more phones on average on a phone with a 5 digit PIN than a 4 digit PIN given that the attacker only gets 5 tries. For most of the other lengths, the same types of patterns and distributions as for 4 digit PINs emerge, with the exception of PINs of 7 digits to some degree. Berry speculated that this was because people instead used their phone numbers without their area codes [6]. These tables are great examples of how attackers would build their dictionary attacks, as some PINs are used way more than others.

| # | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PIN | % | PIN | % | PIN | % | PIN | % | PIN | % | PIN | % |
| 1 | 12345 | 22.802% | 123456 | 11.684% | 1234567 | 3.440% | 12345678 | 11.825% | 123456789 | 35.259 | 1234567890 | 20.431 |
| 2 | 11111 | 4.484% | 123123 | 1.370% | 7777777 | 1.721% | 11111111 | 1.326% | 987654321 | 3.661% | 0123456789 | 2.323% |
| 3 | 55555 | 1.769% | 111111 | 1.296% | 1111111 | 0.637% | 88888888 | 0.959% | 123123123 | 1.587% | 0987654321 | 2.271% |
| 4 | 00000 | 1.258% | 121212 | 0.623% | 8675309 | 0.465% | 87654321 | 0.815% | 789456123 | 1.183% | 1111111111 | 2.087% |
| 5 | 54321 | 1.196% | 123321 | 0.591% | 1234321 | 0.220% | 00000000 | 0.675% | 999999999 | 0.825% | 1029384756 | 1.293% |
| 6 | 13579 | 1.112% | 666666 | 0.0577% | 0000000 | 0.188% | 12341234 | 0.569% | 147258369 | 0.591% | 9876543210 | 0.971% |
| 7 | 77777 | 0.618% | 000000 | 0.521% | 4830033 | 0.158% | 69696969 | 0.348% | 741852963 | 0.455% | 0000000000 | 0.942% |
| 8 | 22222 | 0.454% | 654321 | 0.506% | 7654321 | 0.154% | 12121212 | 0.320% | 111111111 | 0.425% | 1357924680 | 0.479% |
| 9 | 12321 | 0.412% | 696969 | 0.454% | 5201314 | 0.128% | 11223344 | 0.293% | 123454321 | 0.413% | 1122334455 | 0.441% |
| 10 | 99999 | 0.397% | 112233 | 0.417% | 0123456 | 0.124% | 77777777 | 0.275% | 123456789 | 0.378% | 1234512345 | 0.402% |
| 11 | 33333 | 0.338% | 159753 | 0.283% | 2848048 | 0.124% | 77777777 | 0.262% | 147852369 | 0.356% | 1234554321 | 0.380% |
| 12 | 00700 | 0.261% | 292513 | 0.250% | 7005425 | 0.120% | 99999999 | 0.223% | 111222333 | 0.304% | 5555555555 | 0.259% |
| 13 | 90210 | 0.244% | 131313 | 0.235% | 1080413 | 0.111% | 22222222 | 0.219% | 963852741 | 0.255% | 1212121212 | 0.244% |
| 14 | 88888 | 0.217% | 123654 | 0.228% | 7895123 | 0.195% | 55555555 | 0.205% | 321654987 | 0.253% | 9999999999 | 0.231% |
| 15 | 38317 | 0.216% | 222222 | 0.212% | 1869510 | 0.102% | 33333333 | 0.176% | 420420420 | 0.241% | 2222222222 | 0.219% |
| 16 | 09876 | 0.185% | 789456 | 0.209% | 32233326 | 0.100% | 44444444 | 0.165% | 007007007 | 0.227% | 7777777777 | 0.206% |
| 17 | 44444 | 0.179% | 999999 | 0.194% | 1212123 | 0.096% | 66666666 | 0.160% | 135792468 | 0.164% | 3141592654 | 0.195% |
| 18 | 98765 | 0.169% | 101010 | 0.190% | 1478963 | 0.088% | 11112222 | 0.140% | 397029049 | 0.158% | 3333333333 | 0.186% |
| 19 | 01234 | 0.160% | 777777 | 0.188% | 2222222 | 0.085% | 13131313 | 0.131% | 012345678 | 0.154% | 7894561230 | 0.165% |
| 20 | 42069 | 0.154% | 007007 | 0.186% | 5555555 | 0.082% | 10041004 | 0.127% | 123698745 | 0.152% | 1234567891 | 0.161% |

Table 2: Frequency of the top 20 most used PINs for lengths 5-10 [6]

In 2015, a device that could brute-force iPhone PIN code screen lock, even with the "Erase data after 10 attempts" option enabled on the phone. The device was called an IP Box, which sends PIN codes sequentially over the USB connection of the device. The device also connects directly to the phones power source, and cuts the power to the phone after every attempt before the attempt has been saved to flash memory. Each attempt took 40 seconds, so trying all 4 digit PIN codes would take a little over 111 hours or 4.6 days [24].

### 2.2.2 Entropy calculation

In this subsection we will further explain entropy. The National Institute of Standards and Technology (NIST) provides detailed information on the entropy of PINs and passwords in [7]. As stated by NIST, the article is outdated and superseded by other more recent publications, but as the Appendix on password entropy do not exist in any of the more recent publications, we had to use the outdated publication. Even though the publication is outdated we do not believe that the information on password entropy is, making us able to use the article in this subsection.

Claude Shannon coined the term "entropy" for use in information theory, as well as using it to express the amount of actual information in English text. Shannon stated that: "The entropy is a statistical parameter which measures in a certain sense, how much information is produced on the average for each letter of a text in the language. If the language is translated into binary digits (0 or 1) in the most efficient way, the entropy H is the average number of binary digits required per letter of the original language." Here, entropy denotes the uncertainty in the value of a password expressed in bits. As stated earlier, the strongest passwords are those that are truly random, but this is rarely the case for most users, who instead make easier to remember passwords that are not random. Hence, cryptographers have derived a number of alternative forms of entropy, including guessing entropy and min-entropy, which we will describe here.

**Random passwords:**

Measuring the entropy of truly random passwords is done by using the equation

$$H = log_2(b^l)$$

for randomly chosen passwords, the guessing entropy and min-entropy all have the same number of bits. Measuring passwords that are not random as random gives a big false sense of security. For example, the password "Password1" will have an entropy of 53.6 bits if calculated by the equation above, when in reality, "Password1" is a password with much lower entropy.

**Guessing entropy:**

Measuring the entropy of user-chosen passwords is more difficult than measuring the entropy of randomly chosen password, as they do not have a uniform random distribution [7]. User-chosen passwords probably roughly reflect the patterns and character frequency distributions of regular English. As stated earlier, many users will choose passwords that are fairly easily guessable and inside a dictionary of just a few thousand commonly used passwords. Guessing entropy can be said to be the most critical measure of the strength of a password system because it largely reflects the resistance to targeted, online password guessing attacks. In their guidelines, NIST chose to

use Shannon's estimate of the entropy in regular English text as a starting point to estimate the entropy of user-chosen passwords. In his experiments on English text, Shannon used a 27 character alphabet consisting of the English lower case letters and the space. Normally, passwords can be selected from the normal keyboard alphabet of 94 printable characters, but the assumptions, as stated above also, is that left to their own, users will choose passwords consisting of almost entirely lower case letters, and put and upper case letters and number at the start and end of the password respectively. Shannon found that even though there is a non-uniform probability distribution of letters, it is comparatively hard to guess the first letter in a string of English text, but after the first letter it is much easier to guess to second, and after the two first it is easier to guess the third and so on. Table 2.2.2 shows the entropy of both user chosen passwords and PINs from the NIST publication [7]. They state that these values should not be taken as accurate estimates of absolute entropy, but gives a rough relative estimate of the likely entropy of user chosen passwords and PINs. In the list below is the ruleset behind table 2.2.2 drawn from the full keyboard alphabet: [7]

- The entropy of the first character is taken to be 4 bits;
- The entropy of the next 7 characters are 2 bits per character; this is roughly consistent with Shannon's estimate that "when statistical effects extending over not more than 8 letters are considered the entropy is roughly 2.3 bits per character;"
- For the 9th through the 20th character the entropy is taken to be 1.5 bits per character;
- For characters 21 and above the entropy is taken to be 1 bit per character;
- A "bonus" of 6 bits of entropy is assigned for a composition rule that requires both upper case and non-alphabetic characters. This forces the use of these characters, but in many cases these characters will occur only at the beginning or the end of the password, and it reduces the total search space somewhat, so the benefit is probably modest and nearly independent of the length of the password;
- A bonus of up to 6 bits of entropy is added for an extensive dictionary check. If the Attacker knows the dictionary, he can avoid testing those passwords, and will in any event, be able to guess much of the dictionary, which will, however, be the most likely selected passwords in the absence of a dictionary rule. The assumption is that most of the guessing entropy benefits for a dictionary test accrue to relatively short passwords, because any long password that can be remembered must necessarily be a "pass-phrase" composed of dictionary words, so the bonus declines to zero at 20 characters.

For user-chosen PINs the assumption of the above list is that the PIN is subjected to at least a rule that prevents the use of all the same digit and runs of digits (1234 or 98765), though NIST states that this is at best a very crude estimate and that experience with password crackers suggests that many users will often choose simple number patterns and recent dates, for example their year of birth.

**Min-entropy:**

The min-entropy is the measure of the most guessable password in a security system, meaning that an attacker who is determined to find the password of any user on a system would try the most

| | User chosen 94 Character Alphabet | | | Randomly chosen | | |
|---|---|---|---|---|---|---|
| | | | | 10 char. alphabet | | 94 char. alphabet |
| **Length Char.** | **No Checks** | **Dictionary Rule** | **Dict. & Composition Rule** | | | |
| **1** | 4 | - | - | 3 | 3,3 | 6,6 |
| **2** | 6 | - | - | 5 | 6,7 | 13,2 |
| **3** | 8 | - | - | 7 | 10 | 19,8 |
| **4** | 10 | 14 | 16 | 9 | 13,3 | 26,3 |
| **5** | 12 | 17 | 20 | 10 | 16,7 | 32,9 |
| **6** | 14 | 20 | 23 | 11 | 20 | 39,5 |
| **7** | 16 | 22 | 27 | 12 | 23,3 | 46,1 |
| **8** | 18 | 24 | 30 | 13 | 26,6 | 52,7 |
| **10** | 21 | 26 | 32 | 15 | 33,3 | 65,9 |
| **12** | 24 | 28 | 34 | 17 | 40 | 79 |
| **14** | 27 | 30 | 36 | 19 | 46,6 | 92,2 |
| **16** | 30 | 32 | 38 | 21 | 53,3 | 105,4 |
| **18** | 33 | 34 | 40 | 23 | 59,9 | 118,5 |
| **20** | 36 | 36 | 42 | 25 | 66,6 | 131,7 |
| **22** | 38 | 38 | 44 | 27 | 73,3 | 144,7 |
| **24** | 40 | 40 | 46 | 29 | 79,9 | 158 |
| **30** | 46 | 46 | 52 | 35 | 99,9 | 197,2 |
| **40** | 56 | 56 | 62 | 45 | 133,2 | 263,4 |

Table 3: Estimated password guessing entropy in bits vs. password length [7]

probable password with every username, then the second most probable password and so on until the attacker guesses the first correct password for a user [7]. As an example they give a security system in which one user in 1,000 chooses on of the 2 most common passwords and any user user is locked out after 3 unsuccessful authentication attempts. An attacker with a list of user names who knows the most commonly chosen passwords can try those 2 passwords for every user. Doing this, the attacker finds at least one password about half the time by trying 700 usernames. If an attacker is only seeking to get access to any user's account, this is clearly a practical attack.

As there is no way of estimating the actual min-entropy of a security system without examining the passwords in detail with the selected rules of the system, the researchers as NIST could only suggest to run all passwords against a sizeable dictionary of commonly chosen passwords under the system rules. Below is a list of measures to ensure at least 10 bits of min-entropy: [7]

- Upper case letters in passwords are converted to entirely lower case and compared to a dictionary of at least 50,000 commonly selected otherwise legal passwords and rejected if they match any dictionary entry, and
- Passwords that are detectable permutations of the username are not allowed.

## 2.3 Pattern

Pattern authentication was created to be a new way of authentication to oppose PINs and passwords, as humans tend to remember graphical passcodes better than traditional PINs or passwords [25]. To unlock a device with pattern authentication, the user is asked to create a pattern on a 3x3 grid, for example like the pattern in figure 2. When creating a pattern, there are 5 rules [8]:

1. A pattern needs to be at least 4 dots long.
2. A dot can only be used once. The maximum length of the pattern is therefore 9.
3. The pattern will always connect to all dots along a path, except when that dot has already been selected.
4. A pattern can go through previously selected dots to connect dots along the same path.
5. The dots can be connected horizontally, vertically and diagonally.


Figure 2: Pattern authentication method

The maximum number of possible patterns is 389,112. When using less than 9 dots, the number of possible combinations falls rapidly, as shown in table 4. This should in theory offer much better security than the standard 4 digit PIN code, but the researchers in [26] found in their research that typical patterns in reality was as secure as a random 3 digit PIN, thus making pattern authentication less secure than even 4 digit PINs.The research also found that the average pattern length was 5.63. This is about the same as was found in [8], where the average length was found to be 5.40 to 5.92, depending on the application the pattern was used in. During the research in [26] they asked participants to make one defensive and one offensive pattern, and found that 4% of the defensive and 7% of the offensive patterns could be guessed after 10 tries, with these percentages rising to 9% and 19% respectively after 30 tries. They also found that 10% of users use less than 190 different patterns, and less than 300 patterns is used by 50% of the test population.

| Length | Combinations |
|---|---|
| 4 | 1624 |
| 5 | 7152 |
| 6 | 26 016 |
| 7 | 72 912 |
| 8 | 140 704 |
| 9 | 140 704 |
| Total | 389 112 |

Table 4: Number of combinations for different lengths[ [8].

Marte Dybevik Løge showed in [8] that users also tend to create patterns that match a letter of the alphabet, with 11.4% of her collected patterns matching a letter in the English alphabet, with the most popular letters being C, L, small L, M, N, O, S, U and Z.

## 2.4 Fingerprint recognition

Authenticating people with the help of fingerprints have been used for a very long time [27]. It wasn't before the late 19th century that efforts were made to use fingerprints for identification, as this process is more demanding due to due selecting the correct identity of an unknown individual from larger datasets of possible identities, and not only verifying a claimed identity with 1-to-1 comparison [28]. At this point in time, fingerprint recognition is one of the most popular methods for mobile authentication. Most mobile phones today comes with a fingerprint scanner somewhere on the mobile phone. the use of scanners that are placed under the screen of the device is also being released, which opens the possibility of authenticating yourself anywhere on the screen surface of the phone, not only where the small scanner is placed, as on most of today's phones.

| Biometric | Universality | Distinctiveness | Permanence | Collectability | Performance | Acceptability | Circumvention |
|---|---|---|---|---|---|---|---|
| Fingerprint | Medium | High | High | Medium | High | Medium | Medium |

Table 5: Fingerprint characteristics

### 2.4.1 Basics of fingerprint analysis

When analyzing the ridges and valleys of a fingerprint, different patterns and structures of the fingerprint is used. In this subsection, we will describe these patterns and structures, and provide figures showing them. We will start by describing type lines and the two singularities that is the core point and delta point. We will then go into the different kinds of core points, and end with describing the different kinds of minutiae found in [5].

**Pattern Area:**

The pattern area of a fingerprint is defined as the main part of the fingerprint surrounded by the type lines. This is shown in 3.

**Type lines:**

Type lines are the two ridges that starts off parallel on one side of the finger and then diverges. The

type lines define the pattern area and may not be continuous. If there is a break in one of the type line ridges, the nearest ridge lying outside is considered to be the continuation of the old type line. Type lines are shown in 3.

**Core point:**
The core point in a fingerprint is a singular point where the curvature of the ridges reaches a maximum. The core can be considered as a U-turn or the ridge ending enclosed by it in the fingerprint. The core point is meant to give an approximation of the center of a fingerprint image is. The core is shown in figure 3.

**Delta point:**
The other type of singularity is called a delta point. A delta point is a place in the fingerprint where two ridges run side-by-side and then diverges with a significant pattern area in front of the divergence. Simplified, this means that a triangle is formed. If there are multiple deltas on the same fingerprint, the nearest ridge point to the center of the divergence of type lines is taken as the right delta to use for further analysis and comparison. If there are multiple points open toward the core at the point of divergence of the two type lines the closest core is picked as the delta point of the fingerprint. Figure 3 points to one of the two delta points for that fingerprint.

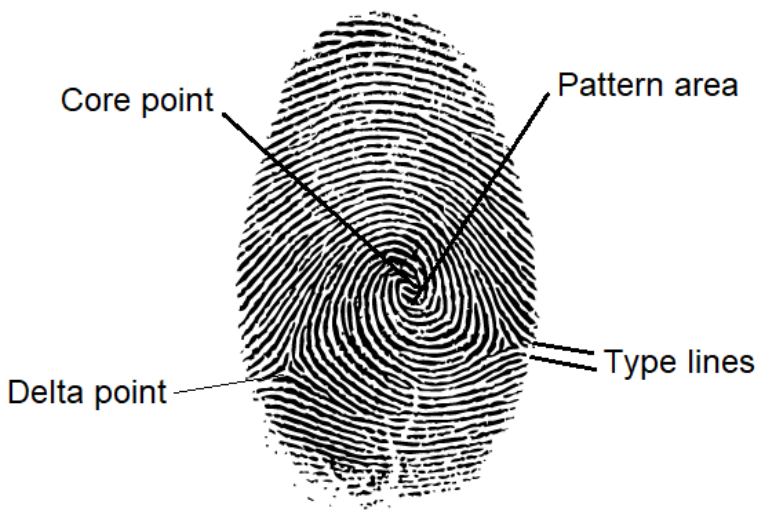


Figure 3: Pattern area, type lines, core point and delta point in a fingerprint. Adapted from [1] ©Dusi Puffi/Adobe Stock

There are 3 main classifications of core points, with some having sub classes further dividing the patterns. The three main classifications are arch patterns, loop patterns and whorl patterns [5]:

**Arch patterns:**
A pattern in a fingerprint is classified as an arch pattern when there are convex lines flowing from one side of the finger to the other without any significant singularities. There are two sub classes of

arch patterns, these are [5]:

- **Plain arch**
  A plain arch pattern is made of ridges that goes from side of the finger to the other with only a slight waveform. The plain arch pattern doesn't have a delta nor a significant core. Pattern is shown in figure 4.
- **Tented arch**
  The tented arch pattern is similar to the plain arch, with some differences. There is three kinds of tented arches:
  1. Ridges in the center of the fingerprint form a definitive angle.
  2. Arch with a heavy upthrust in the center.
  3. An arch nearly being under the loop classification but missing one of the essential characteristics of that classification. If the fingerprint has a delta but not a sufficient recurve is also classified as a tented arch.

Figure 4: Arch pattern [2]. ©Kevin Chesson/Adobe Stock

**Loop patterns:**
A fingerprint is classified as a loop when one or more of the ridges the fingerprint enters from one side, curves and goes out on the same side. A loop pattern can either be a right loop or a left loop depending on where the ridge lines enters and exits. If the ridges enter and exits on the left side of the finger, the pattern is a left loop, if they enter and exit on the right side of the finger, the pattern is i right loop. For left loops, the delta in the fingerprint is on the right side, and for right loops, the delta is on the left side . Loop patterns are shown in figure 5 [5].

Figure 5: Right loop. Adapted from [3]. ©chege/Adobe Stock

**Whorl patterns:**
A fingerprint pattern is classified as a whorl when the ridges form a revolution around the center of the fingerprint. Whorl patterns has at least two deltas, global convex ridges and at least one ridge that creates a full circle, making an overall circular effect of the pattern. As with arches, whorls also have sub classes. These are plain whorl, central pocket loop, double loop and and accidental whorl [5].

- **Plain whorl**
  The plain whorl pattern has two deltas and at least one ridge that makes u full circle. If an imaginary line is drawn between the two deltas, one of the recurving ridges has to touch the imaginary line within the pattern area [5].
- **Central pocket loop**
  The central pocket loop pattern has two deltas and at least one ridge that makes a full circle. If an imaginary line is drawn between the two deltas, the line can not touch any of the recurving ridges in the pattern area [5].
- **Double loop**
  The double loop pattern is made of two loop patterns with two independent and distinct shoulders and and deltas [5].
- **Accidental whorl**
  The accidental whorl pattern is also often called composite, and is an uncommon pattern to

occur in a fingerprint. The pattern consists of two different kinds of patterns, and has more than two deltas [5].

Figure 6 shows a plain whorl pattern.

Figure 6: Plain whorl [4]. ©Jashin/Adobe Stock

When identifying fingerprints, minutiae, also called Galton details are the most important marks in the fingerprint. Minutiae points are classified into different classes. The most usual classes are shown in figure 7 and described below [5].

- **Ridge ending**
  A ridge ending is where a ridge begins or ends abruptly. This ridge can be either long or short.
- **Bifurcation**
  A bifurcation is when a ridge splits into two parallel ridges.
- **Trifurcation**
  A trifurcation is when a ridge splits into three parallel ridges.
- **Divergence**
  A divergence is when two ridges that run alongside each other spreads.
- **Lake**
  A lake is when a ridge splits into two, forming a bifurcation, and then splicing back together (forming another bifurcation). This leaves a valley in between the two bifurcations, which can look like a lake.
- **Independent ridge**
  An independent ridge is a short ridge independent from other ridges.
- **Spur**
  A spur is similar to a bifurcation, but where one of the branching ridges is smaller than the other.
- **Crossover**

A crossover is when an independent ridge span from one ridge to another parallel ridge.

- **Line brake**
  A line break is when a ridge suddenly stops, then starts again, creating two line endings close to each other.

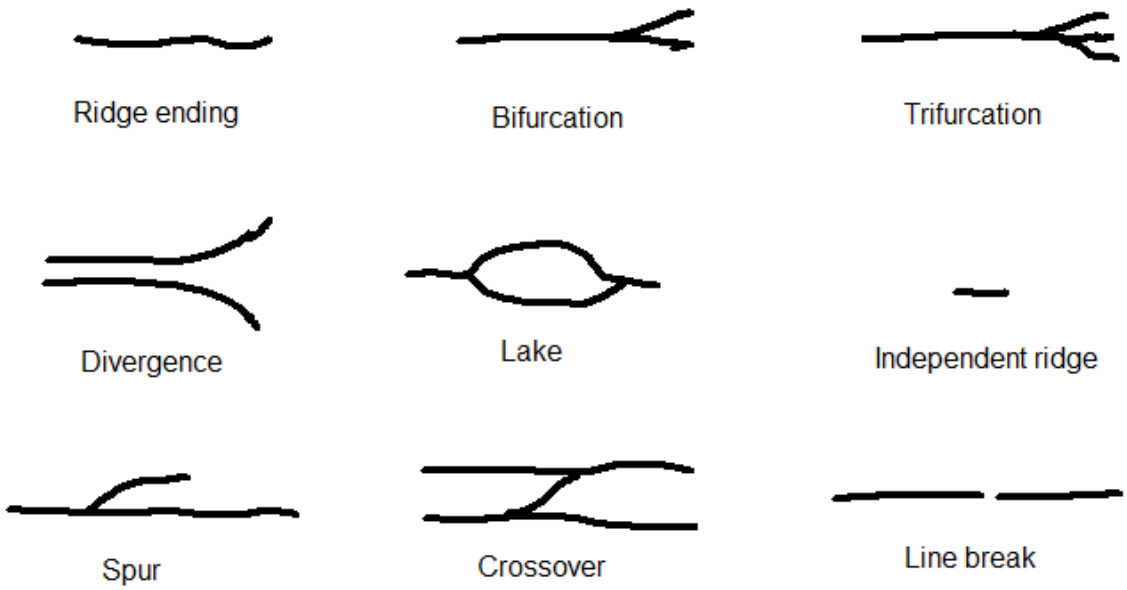


Figure 7: Galton details of a fingerprint. Adapted from [5]

For automatic comparison of fingerprints, only ridge endings and bifurcations along with their directions are used. For manual comparison, all the above classifications as well as others can be used. The reason only ridge endings and bifurcations are used in automatic comparison is that most of the other classifications are variations or of these two classes, or patterns derived from these two classes, and using more classes can slow down comparison for an automatic system.

### 2.4.2 Attacks

Attacks on biometric systems can, according to [29] be executed using eight different methods. These methods are:

1. **Presenting a fake biometric probe**
   This method of attack is performed by presenting a fake biometric probe to the sensor. This could be a fake finger made of for example gelatin, or presenting a picture, video or face mask to a facial recognition system.
2. **Resubmitting previously stored biometric data**
   This method of attack is performed by resending previously recorded digitized signal to the

system while bypassing the sensor. This could be done by for example presenting an old image of a fingerprint image.

3. **Overriding the feature extraction**
   This method uses a trojan to attack the feature extractor and make it produce the feature sets the attacker wants.
4. **Tampering with the feature representation**
   This method of attack is performed by replacing the features extracted with a fake feature set (assuming the representation method is known).
5. **Compromising the matcher**
   The matcher is attacked and compromised so it produces a predetermined match score.
6. **Tampering with the stored templates**
   If an attacker can break into the database and modify templates, he can authenticate an impostor or falsely reject an individual associated with the template.
7. **Attacking the channel between stored templates and the matcher**
   When the stored templates are sent to the matcher, the data could be intercepted and altered.
8. **Overriding the final decision**
   If an attacker can override the final decision the system is compromised and disabled. Even if the system has perfect performance characteristics it's been rendered useless by overriding any result from the matcher.

In this literature review we will focus mostly on presentation attacks and presentation attack detection, though the next subsection will contain some measures to resist all the types of attack. Since the types of attack are equal to all biometric security systems, they will not be listed and explained in the following sections about biometric security systems.

As the fingerprint sensor in most mobile phones are smaller than normal sensors, thus capturing a smaller area of the fingerprint and capturing less minutiae. To compensate for the small sensor in mobile phones, multiple partial prints are collected for the same finger during enrolment, and the user is able to enroll multiple fingers to further enhance the usability in the system. The authentication is also unsupervised in that the system does not know which finger or which part of a finger is being placed on the sensor. Because of this, the system can declare an authentication attempt successful if the sensed fingerprint matches with any of the stored partial fingerprints of any enrolled finger [30]. The work of Roy, Memon and Ross in [30] exposed a vulnerability in fingerprint authentication systems that only use partial fingerprints, like mobile phones. They found that it was possible to create what they call a MasterPrint to launch a dictionary attack against the system. The experiment was conducted using the FVC2002 DB1-A and FingerPass DB7 datasets, and came up with two approaches to generate these MasterPrints. The first method generated a MasterPrint using a dataset, while the other method generated its MasterPrint synthetically using a hill climbing method. With a dictionary attack of 5 partial fingerprint based MasterPrints, with a theoretical 5 attempt maximum to be authenticated it was possible to attack 26.46% of users (12 impressions per finger) for the FingerPass DB7 capacitive dataset and 65.20% of users (average

80 partial impressions per finger) in the FVC optical dataset when the FMR was set to 0.1%. They also found that the synthetic MasterPrints performed better than the sampled MasterPrints, and that even if a MasterPrint only matches with a small number of partial fingerprints, the number of subjects that it matches against can be high due to each subject storing multiple partial fingerprints. At a FMR of 0.1% a single MasterPrint from the FingerPass DB7 dataset only matched to 1.4% of the partial prints, but this corresponded to 10.6% of subjects in the dataset since every subject had 12 impressions. Enrolling multiple fingers to the same system will only increase the risk that a MasterPrint will match one of the impressions of a user.

Another common attack against fingerprint and other biometric systems is presentation attacks, where the attacker presents a fake biometric probe to the sensor. The type of sensor determines what kind of materials that can be used to fool it. For example, a capacitive fingerprint sensor will only read a fake fingerprint if it is conductive, and only match if it is of similar conductivity as a human finger. We ourselves tried one such attack described in section 3.1, with our attack being mostly unsuccessful. In the start of April 2019, a man called darkshark posted an attack against the new Samsung Galaxy S10 on the website Imgur, where he showed himself unlocking his Galaxy S10 with a 3D printed fingerprint he collected via the same phones camera. After the image was taken he put it into Photoshop and increased the contrast, and and created an alpha mask. Then, the image was converted to a 3D model and then 3D printed. The video shows him unlocking the phone effortlessly with the newly printed fingerprint [31].

In 2104, a hacker named Jan Krissler (known as Starbug in hacker circles) created a presumably working fake fingerprint from high resolution photographs of Germany's defence minister Ursula von der Leyen. The photographs were obtained by photographing her hand at a press conference, as well as making use of a photograph released by a government press office [32, 33]. A translated video of the originally German demonstration is available in [34]. He also spoofed the fingerprint sensor of the iPhone 5S in less then a day when the phone came out in 2013 by lifting a fingerprint from the iPhones screen, though this was done in cooperation with the "victim" of the attack [32].

### 2.4.3 Presentation attack detection

Implementing measures to resist against attacks is vital in all security system, biometric or otherwise. The techniques described in this subsection will be applicable to all biometric security systems, though we will use fingerprint authentication as examples in this subsection. We will also focus on presentation attack detection, but will explain measures for other types of attack as well, as we stated in the previous subsection.

According to [35] there are many ways to protect a biometric security system. We will mostly focus on protecting against presentation attacks, but in this subsection we will describe some other measures that can be taken to protect the system as well.

- **Liveness detection**
  Liveness detection is a mechanism that can be implemented to check if the presented finger is real and provided by a living person or not, and is used to prevent attacks against the sensor. Liveness detection can be implemented via hardware or software. Liveness detection imple-

mented with hardware can be built to measure different life signs like pulse detection, blood pressure, finger temperature and more. The limitation of liveness detection implemented with hardware is the extra cost, and especially for mobile devices, the extra space needed for these extra sensors. Liveness detection implemented in software means using features already captured by the sensor that captures the fingerprint. The only method of that today is using sweat pore information, which requires a high-resolution scanner, as recreating the size and position of sweat pores in a fingerprint on a artificially made fingerprint is extremely hard to do.

- **Biometric cryptosystems**
  Combining biometrics and cryptography is done to take the advantages of both fields. Cryptography ensures better security while the use of biometrics eliminates the need to remember passwords or carry tokens. Traditionally, cryptographic systems convert plain text into cipher text using one or multiple keys. To decrypt the cipher text into plain text, the corresponding decryption keys are needed, and without them an attacker will not be able to derive any meaning from the cipher text. For biometric cryptosystems, the process is divided into two subdivisions, the key generation and key binding:
  - **Key generation**
    Helper data is obtained from biometric traits and the cryptographic key is directly generated from said helper data.
  - **Key release**
    Helper data is obtained by binding a key with a biometric template.
- **Steganography and watermarking**
  Steganography is derived from Greek and means "covered writing", and in digital systems refers to the act of hiding information inside other data. A good example of this is hiding information in the least significant bits of an image. Changing only the least significant bits in the different pixels of the image will not change it enough that it is perceptible without comparing the two images side by side and by using extra tools. Sending an image over the net isn't often seen as suspicious, effectively hiding the actual information that is sent, and this is what is used when images are watermarked. Steganography and watermarking can be used to stop attacks on the channel between the sensor and feature extractor and on the channel between the matcher and application device. This is done by using watermarking for ensuring ownership claim when authenticating while steganography can be used for transferring critical biometric information from a client to a server.
- **Cancellable biometrics**
  Cancellable biometrics is a technique that is used to be able to revoke and more securely store biometric templates. Since biometric features don't change much or at all over time, having the ability to revoke or remake biometric templates is more or less necessary . This means that if an attacker gets their hands on a database of biometric templates, these templates won't be of any use on other biometric security systems, and not even on the breached system after the biometric templates have been revoked or remade. To do this, the biometric templates

are purposefully and systematically distorted on a selected non-invertible transform before being stored. If a template is stolen, a new one can simply be created by changing the parameters of the non-invertible transform. This kind of measure is taken against attacks on the template database, while also addressing the problem of non-replaceability that is a natural characteristic of biometrics.

### 2.4.4 Performance

The performance of fingerprint recognition systems is well researched, though often for full sized sensors. Research on the performance of small sensors such as those in mobile phones do exist, but the found performance is not great, with equal error rates (EER) ranging from approximately 11-17% for sensors of 10x10 mm, while 8x8 mm sensors have an EER of approximately 19-32%. While the false match rate (FMR) and false non-match rate (FNMR) can be altered with thresholds, having a high EER is indicative of a system that does not perform very well. In the specific case of Apple's Touch ID they have claimed that the chance of a random person unlocking your phone with their fingerprint is one in 50,000 when one finger is enrolled, decreasing to 1 in 10,000 for five enrolled fingers [36]. This would constitute a FAR of 0.002% for one finger enrolment, which would be more secure than a random 4 digit PIN. Finding specific numbers from other manufacturers have proved difficult, so we will from this point use this FAR value in any calculations for fingerprint recognition in other chapters.

## 2.5 Facial recognition

Using facial recognition for authentication is gaining popularity on mobile phones, with the newest generations of iPhones not even offering any other biometric authentication method except for their Face ID. Facial recognition is seen as having a high acceptability among the population, as well as being able to capture data without touching the a sensor. Not touching a sensor is not as important in mobile phone authentication, as the sensor is embedded in the phone of the owner, and the owner shouldn't have a problem touching his or hers own phone. The implementation of facial recognition between mobile phone brands and their generations of phones differ, as some implements 2D recognition, while other implements the more secure 3D recognition. While 3D facial recognition is more secure and less susceptible to simple presentation attacks, it does require an extra sensor other than just the front facing camera, like on the newest iPhones where a infrared camera projects and reads over 30,000 infrared dots to form a depth map of the users face [36]. Table 2.5 and 2.5 shows the 7 characteristics of both 2D and 3D facial recognition. The characteristics is mostly alike, except for distinctiveness and circumvention. 3D recognition scores better in both these characteristics as adding depth increases the distinctiveness of faces, while also making presentation attacks harder for the attacker.

| Biometric | Universality | Distinctiveness | Permanence | Collectability | Performance | Acceptability | Circumvention |
|---|---|---|---|---|---|---|---|
| 2D facial rec. | High | Low | Medium | High | Low | High | High |

Table 6: 2D facial recognition characteristics

| Biometric | Universality | Distinctiveness | Permanence | Collectability | Performance | Acceptability | Circumvention |
|---|---|---|---|---|---|---|---|
| 3D facial rec. | High | Medium | Medium | High | High | High | Low |

Table 7: 3D facial recognition characteristics [9]

To our knowledge, the best implementation of facial recognition in phones available today belongs to Apple's iPhone X and XS series using Face ID. These phones utilizes the TrueDepth camera for 3D recognition, that looks for your face when it is woken from being locked by raising the phone or tapping the screen. When a face is detected, Face ID confirms attention and intent by the user by checking if the user is looking at the screen with their eyes open. When attention is confirmed, the TrueDepth camera projects and reads over 30,000 infrared dots that forms a depth map of the face, along with a 2D infrared image. This data is used to create a sequence of depth maps and 2d images, that then gets digitally signed and sent to the Secure Enclave, which provides all cryptographic operations for data protection key management and maintains the integrity of data protection even when the kernel has been compromised. To prevent digital and physical spoofs of the data, the TrueDepth camera randomizes the sequence of depth map and 2d image captures and projects it in a pattern that is specific to each device. Protected by the Secure Enclave, a portion of the iPhone's A11 processor's and newer system on chips (SoC) neural engine transform the data into a mathematical representation and then compares it to the many poses collected during the enrolment stage. To better this process, neural networks specifically trained for facial matching with a training set of over a billion images including infrared and depth images collected in Apple studies is used. These neural networks are trained with data ranging a wide variety of age, gender, ethnicity and more, as well as there being a neural network trained to resist against presentation attacks with photos and masks. The Face ID data will also never leave the phone, is not included in device backups and always stay encrypted inside the Secure Enclave, with the data also only being available to the Secure Enclave. [36].

### 2.5.1 Basics of facial recognition

A facial recognition system is usually made of the four modules face localization, normalization, feature extraction and matching, further described below [37]:

**Face localization and landmarking**

Face localization detects and segments the face area from the background of the image. While it provides an estimate of the location and scale of the face in the image, face landmarking finds facial landmarks like the eyes, nose mouth and facial outline. This can be accomplished by using a landmarking module or face alignment module.

**Face normalization**

Face normalization normalizes the face geometrically and photometrically. This is done so the system can recognize users under varying poses and lighting. The geometrical normalization process transforms the face into a standard frame by cropping the face, and may also warp or morph for more elaborate geometric normalization. The photometric normalization process normalizes the face based on properties such as lighting and gray scale.

**Feature extraction**

In the feature extraction module the useful features for distinguishing faces of different people is extracted from the normalized face. The features must be robust for differentiating faces with respect to the geometric and photometric variations, and are used for matching.

**Matching**

In the matching module, the extracted features from the normalized face is matched against the enrolled face or faces in the database. In the case of authentication, the matcher outputs either a yes or no. The challenge in this module is to find a suitable similarity metric for comparing facial features. The performance of the security system is highly dependent on the features and their quality that are extracted, which again is dependent on the face localization and normalization modules.

### 2.5.2 Attacks

Presentation attacks against facial recognition systems usually involves displaying a photo, video or a mask in front of the sensor. Which attacks that will work or not depends on the implementation and the presentation attack detection. Phones using 2D facial recognition will be susceptible to all these kinds of attacks, which has been documented well with the newest Samsung Galaxy S10, where multiple people have shown that they are able to unlock their phone with a photo or video of themselves [38, 39, 40]. We ourselves also managed to do this, described in section 3.1. Since it is possible to get these images from the internet, using the facial recognition feature on phones with 2D facial recognition like the Galaxy S10 is not secure, and should not be used in any kind of secure setting. The attack on the Galaxy S10 works because the phone can't tell that there isn't an actual human being presenting their face to the camera, only that what it sees from the camera matches what is stored in the template.

Having a mobile phone with 3D facial recognition like the newest iPhones should resist attacks from photos and videos, as these attacks presents 2D images to a 3D sensor. The sensor will see that there is no depth to the presented face, thus disregard it entirely. To fool a 3D sensor, a mask is needed so the depth of the fake face is retained. Making such a mask requires both more time, effort and money, which can deter some attackers. Getting the required data to make such a mask is also harder and requires more planning than taking a single photo of the victim, or potentially just downloading said photo from one of the victim's social media accounts. For determined attackers on the other hand, though harder, making such a mask is not is not that hard. According to Bkav, a mobile phone and cyber security firm in Vietnam, making such a mask can be done by setting up cameras at specific angles pointed at for example a doorway to catch the victim more easily. These photos are then processed by algorithms to make a 3D object. Bkav also made their own mask trying to fool the the iPhone X. Their mask was made using stone powder with 2D infrared images of the eyes glued to the mask. According to bkav, such a mask would cost around 200 USD make [41].

Because you can make such a mask from photographs taken by a setup of cameras, it would not be a stretch to think that it would also be possible to create a 3D object from photographs from a victim's social media. This is exactly what the researchers in [42] showed that they were able to do.

With their method, they managed to fool 4 different commercial authentication systems with a 55% to 85% success rate dependent on the authentication system. Their main goal was circumvention of the presentation attack detection methods implemented in these commercial systems, and was, as far as we can tell against a 2D sensor. Though the attack was not tried against a 3D sensor, the fact the the researchers was successful in creating 3D masks fooling systems with presentation attack detection with photos taken from social media could potentially also pose a threat to 3D sensors, as Bkav showed what they were able to do in their experiments with Face ID.

### 2.5.3 Presentation attack detection

Because 2D facial recognition is so easy to fool, implementing presentation attack detection (PAD) to the security system is important. Because of what we've seen with the attack to the Galaxy S10, these presentation attack detection mechanisms is at best implemented to a varying degree across the landscape of mobile phones using 2D recognition. This might be due to the computational cost of the measures, as users tend to choose easy the most convenient authentication methods. If an authentication method suddenly takes as little as 0.5-1 second longer than it did before to authenticate a user, the user experience may go down drastically. Table 2.5.3 shows a summary of published 2D facial recognition PAD put together by the researchers in [10]

| Method | Strength | Limitation | State of the art performance† |
|---|---|---|---|
| Face motion analysis | Effective for print attack | Requires multiple frames, Slow response | ZJU Eyeblink [43] (Intra-DB, Cross-DB)<br>[43]: (95.7%, n/a)<br>Idiap Replay-attack [44] (Intra-DB, Cross-DB)<br>[45]: (1.25%, n/a); [46]: (0.00%, n/a)<br>CASIA FASD (Intra-DB, Cross-DB)<br>[46]: (21.75%, n/a) |
| Face texture analysis | Relatively low computational cost and fast response | Poor generelizability, Requires face and/or landmark detection | Idiap Replay-attack (Intra-DB, Cross-DB)<br>[47]: (15.54%, 47.1%); [48]: (0.8%, n/a)<br>[49]: (2.9%, 16.7%); [50]: (1.0%, n/a)<br>CASIA FASD (Intra-DB, Cross-DB)<br>[49]: (6.2%, 37.6%); [50]: (7.2%, 30.2% EER) |
| Face 3D shape or depth analysis | Effective for 2D attacks | Requires multiple frames or additional devices | Idiap Replay-attack [51] (Intra-DB, Cross-DB)<br>[51]: (12.5%, n/a) |
| Image quality analysis | Good generalizability,<br>Low computational cost,<br>Fast response time,<br>Face and/or landmark detection not required | Image quality measures can be device dependent | Idiap Replay-attack (Intra-DB, Cross-DB)<br>[52]: (7.41%, 26.9%); [53]: (15.2%, n/a)<br>CASIA FASD (Intra-DB, Cross-DB)<br>[52]: (12.9%, 43.7%)<br>MSU MFSD [52] (Intra-DB, Cross-DB)<br>[52]: (5.82%, 22.6%) |
| Frequency domain anaysis | Good generalization ability,<br>Low computational cost | Spectral features can be device dependent | Idiap Replay-attack (Intra-DB, Cross-DB)<br>[54]: (2.8%, 34.4%)<br>CASIA FASD (Intra-DB, Cross-DB)<br>[54]: (14.0%, 38.5%)<br>UVAD [54] (Intra-DB, Cross-DB)<br>[54]: (29.9%, 40.1%) |
| Active approach | Good generalizability | Requires additional devices | Private 3D mask dataset [55] (Intra-DB, Cross-DB)<br>[55]: (100.0%, n/a) |
| Multi-cue fusion | Good generalizability,<br>Less sensitive to face and/or landmark detection errors,<br>Whole image frame analysis | Moderate computational cost (0.21 sec. on desktop | Idiap Replay-attack (Intra-DB, Cross-DB)<br>Proposed: (0.0%, 3.5%)* (14.6%, 29.3%)**<br>CASIA FASD (Intra-DB, Cross-DB)<br>Proposed: (1.67%, 2.5%)* (5.88%, 35.4%)**<br>MSU MFSD (Intra-DB, Cross-DB)<br>Proposed: (2.67%, 9.27%)* (8.41%, 26.7%)**<br>MSU USSA (Intra-DB, Cross-DB)<br>Proposed: (3.84%, 31.4%) |

†Intra-database results for the Idiap Replay-Attack and UVAD databases are given in terms of Half Total Error Rate (HTER). HTER is defined as the average of false acceptance rate and false rejection. Intra-database results for the CASIA FASD, MSU MFSD and the MSU USSA databases are given in terms of Equal Error Rate (EER). For the ZJU Eyeblink and the Private 3D mask datasets, classification accuracy are reported. Cross-database results are given in terms of HTER unless otherwise specified. *Performance was reported using the proposed smartphone protocol; no reject option was used. **Performance was reported using the original intra-database protocol for each database.

Table 8: Summary of published methods on 2D face presentation attack detection [10]

Though 3D sensors are harder to fool than their 2D counterparts, presentation attack detection such methods such as liveness detection, or any method in table 2.5.3 would increase the security of the system. The same pitfalls are prevalent for 3D recognition as for 2D, where having a system that is not responsive or fast enough will have users find an alternate way to authenticate, with this method often being less secure

### 2.5.4 Performance

As for fingerprint recognition, finding the performance of facial recognition systems specific to different phone brands or generations have proved difficult except for from Apple. We did not manage to find such a document issued by another big phone manufacturer like Samsung, though they do state at their website that using 2D facial recognition on their phones is less secure than pattern, PIN, iris or fingerprint [56]. As Samsung must be aware of, most people use a 4 digit PIN (almost 63% of respondents in our questionnaire, shown in section 3), having a worse performance

than than a 4 digit PIN can not be considered secure by any means, and should not be used in any environment where security is even slightly important.

In the article speaking on the security of iPhones, Apple state that the chance of a random person could unlock your iPhone is one in 1,000,000 [36]. When two faces is enrolled this probability increases to one in 500,000. This constitutes a FAR of 0.0001% and 0.0002% respectively. They also state that the false accept rate is different for twins and siblings who look alike, as well as for children under the age of 13 because their distinct facial features may not be fully developed yet [36]. This is a negative side to facial recognition compared to fingerprint and iris recognition, in that there is a potential for close family and especially identical twins to fool the security system. This problem is not present in fingerprint nor iris recognition as the creation of these patterns is not linked to genes, but are random patterns created in the womb [57, 58]. Thus, the chance of an identical twin fooling the system is greater using facial recognition, and might not be suitable in situations where keeping out close family is the goal. For more professional use, when the goal is to keep out other adversaries, the problem of close family getting into the phone might not be as prevalent, though it can still be seen as a threat that the close family of the users of the system can be able to unlock the phone.

## 2.6 Iris Recognition

Not many mobile phones on the market today includes an option for using iris recognition. The most notable phones that implemented iris recognition was the Samsung Galaxy S8 and S9 series along with the Note8 and Note9, where iris recognition was offered either alone or paired with facial recognition to make authentication more secure, though this option was only on the S9/S9+ and Note9 [59]. The option to use iris recognition for authentication was then dropped in the next generation Galaxy S10 series, reportedly to better the screen-to-body ratio [60].

### 2.6.1 Basics of iris recognition

The iris in itself is about 11 $\pm$ 1.5 mm in diameter and is located behind the cornea. A normal iris recognition system is made up of the 5 modules image acquisition, iris segmentation, quality estimation, feature extraction and matching, further described below [61]:

**Image acquisition**

In the image acquisition module, the user's iris image is obtained. Normally, iris recognition sensors use neat infrared light (NIR) in these sensors, as it better reveals the detailed structure of an individual's iris than visible light does. This is especially true for people with darker eye color.

**Iris segmentation**

When the image is obtained, the iris segmentation module finds and extracts the iris region of the image. In Daugman's method, the inner (pupillary boundary) and outer (limbic boundary) boundaries of the iris is approximated by circles and finding their parameters. Each boundary is found for an image by integro-differential operators, which find the parameters having the highest blurred partial derivatives in a circular arc. After finding the boundaries, the eyelids and eye lashes that are in the iris region is removed. Because the iris chances based on lighting conditions, the

template has to be be prepared for this, and is mapped to a normalized coordinate system, where the pupillary boundary is set to 0 and the limbic boundary is set to 1. This means that any pixel in the new coordinate system is defined by an angle between 0 and 360°and a radial coordinate from 0 to 1.

**Quality estimation**

In the quality estimation module, the system decides if the acquired iris image is of satisfactory quality to proceed with the process or if the image should be discarded. For the measure of quality, Fourier coefficients can be used. A blurry image will not have high-frequency components and will therefore have lower energy in those coefficients, indicating a blurry image.

**Feature extraction**

If the iris image is deemed as accepted, the feature extraction module will compute the features from the iris region. The texture patterns on the iris contains information unique to each person in the world, and to capture this, the normalized iris region is convolved with two dimensional Gabor filters.

**Matching**

When matching, normalized Hamming distance is used to measure the similarity between the input and the template, with the distance only being calculated for those pixels in the images for which both the iris codes are not obstructed. To counteract the in-plane rotation, the distance is computed for multiple rotations of the iris vector, where the lowest distance score is used in the final similarity score.

### 2.6.2 Attacks

As any other system, iris recognition systems are subject to attacks. When the Samsung Galaxy S8 came to market, the same hacker that broke into the an iPhone using a fake fingerprint, also managed to fool that phone's iris recognition. To fool the sensor, an image was taken from medium distance by a camera set to night mode. It was set to night mode as the iris scanner on the phone works with infrared light. The image was then zoomed in on one eye and crop, and then printed out with a laser printer. The last step was placing a contact lens on the printed eye, and presented to the phone. All this is shown in the video in [62].

## 2.7 User-friendliness

User-friendliness consists of four elements: ease of use, naturalness, ease of understanding and helpfulness [63]. User-friendliness can be very subjective, as it may depend on previous knowledge and what each person think is the best. An example of this can be that a young person and an old person might answer differently when they rate a fingerprint recognition system since a young person has grown up with electronics, while an old person might not be as familiar with these devices. The same goes for passwords, where someone might not have a problem with remembering their passwords while others have problems remembering even one or two passwords. Tokens, by itself are usually user friendly as long as the sensor and token is quick and reliable, as a token by itself doesn't require the user to remember anything else than bringing their token. How user-

friendly each authentication method for mobile phones are according to the responses collected via our questionnaire is found in section 3.

## 2.8 Ranking authentication methods

Since our goal with this thesis is finding the best authentication method for mobile phone authentication when considering both security and user-friendliness, it would be helpful for us to look to other similar research. To our knowledge, not much such research has been done, though Helkala and Snekkenes [11] came up with a framework that resembles parts of what we try to accomplish with this thesis. Their framework is built on 4 steps. During each step, every authentication method that does not comply with the steps demands get dropped. The remaining methods moves on to the next step. The four steps of the framework is described as follows:

1. **User and environment compatibility**
   Every authentication method have to comply with usage and environment related requirements. For example, is the users of the security system do a lot of manual labor, fingerprint authentication might not be suitable as their fingerprints might be worn down, making the performance of the system much worse.
2. **Security level compatibility**
   Every remaining authentication method has to meet the set security level. They define the security level as not just the performance or entropy of a security system, but include attacks on both the technical and human factors of the security system. Their framework uses 6 security levels, from no security to extreme security. As their paper was published in 2009, the security levels might be somewhat outdated due to the increase in computational power and the fact that attackers get more and more information to base their dictionary attacks on as time goes on.
3. **Usability**
   After the security level compatibility step, the remaining authentication methods gets checked if it has the usability requirements needed. A practical authentication method should be quick to use and not increase the workload of the end user, as we stated in section 2.7, though they ultimately define the usability as the estimated annual time consumption per user for every authentication method. This annual time is calculated as follows:

   $$Time(a_i) = T(enrol) + t(trans) + t(renew) + t(delayhum) + t(delaysys)$$

   where the summands are time required for enrolment, identity verification, renewing authenticator and delayed transaction times when there is a human or system failure. To be able to compute the annual time for every authentication method, the number of times and and the duration for each authentication attempt needs to be estimated for every authentication method.
4. **Costs**
   The last part of the framework examines the cost of infrastructure and administration for each

authentication method. O'Gorman suggested in [64] how to compute the infrastructure and administration cost. These suggestions are what Helkala and Snekkenes used in their work: the infrastructure cost can be calculated as follows:

$$Info(a_i) = c(equip) + c(soft) + c(imp) + c(ins) + c(enrol) + c(stor)$$

where the sum is the cost of equipment, software, implementation, installation, first enrolment and template storage. When a second enrolment is necessary, the cost is considered to be a cost of administration. Computing the administration cost is done as follows:

$$Adm(a_i) = c(enrol) + c(renew) + c(term) + c(lis) + c(main)$$

where the sum is the cost of user enrolment, authenticator renewal, termination of account and authenticator, software licence and equipment maintenance. The total cost of an authentication method will then be computed by adding the the cost of infrastructure and administration:

$$Cost(a_i) = Inf(a_i) + Adm(a_i)$$

The authentication methods are then ranked, sorted from the lowest cost at the top, to the highest cost at the bottom.

## 2.9 Multifactor authentication

More and more systems are implementing two-factor or multifactor authentication. This takes two or more different authentication mechanisms and combines them to increase the certainty that the user is who he says he is. The most usual implementations of two-factor authentication is combining a token with a password/PIN. Other valuable two-factor or multifactor authentication methods can be password + keystroke dynamics, token + biometric modality and the use of multibiometrics, further described below. Multifactor authentication makes it much harder for adversaries to gain access to a system as they now need a combination of two or more different methods to gain access to a system.

### 2.9.1 Multibiometrics

There are five different approaches to multibiometrics [65]:

1. **Multi-Modal**
   Uses multiple different biometric modalities, for example face+iris.
2. **Multi-Algorithmic**
   Uses one biometric modality, but two or more different algorithms,and combines their comparison scores efficiently to determine if the user gets access.
3. **Multi-Instance**
   During enrollment multiple instances of the same biometric characteristic is captured, for example all ten fingers. For the verification process, all ten fingers are captured once more, and compared to their respective reference.

4. **Multi-Sensorial**
   One biometric characteristic, with multiple capture devices. Example for fingerprints are one capacitive sensor, one optical sensor and one ultrasound sensor.
5. **Multi-Presentation**

   - **Multiple Reference**
     At enrolment one biometric characteristic is recorded multiple time, which makes multiple references. For recognition the biometric characteristic are recorded once, and compared against all the references created at enrolment.

   - **Multiple attempt**
     At enrolment the characteristic is recorded once, and for the recognition it is recorded multiple times and compared to the reference.

Both multi-modal and multi-algorithmic systems can be well suited for biometric authentication in mobile devices, as most devices that offer biometric authentication offer a choice between fingerprint and face, with some also offering iris or other modalities. Mobile devices are also getting fast enough to deploy multiple algorithms to make the authentication system multi-algorithmic, but to our knowledge no such system is deployed. A multi-presentation system with multiple references are already deployed in mobile device biometric authentication systems as all these systems guides you to move your finger around the sensor when making a new template. This is done because the sensor usually is so small that it can't capture the full fingerprint in one capture. Multiple references upon enrolment gives the user more wiggle room when authenticating themselves, as they don't have to hit the exact same spot on the finger for it to find a match.

# 3 Results

## 3.1 Practical attacks

In this section we will perform and describe attacks that we have performed ourselves. The attacks we tried targeted fingerprint and facial recognition sensors on mobile phones.

### 3.1.1 Gelatin fake fingerprint attack

With this attack we tried to break into a Samsung Galaxy S7 edge phone with a fake fingerprint made of gelatin and the Samsung Galaxy S10. The fingerprint mold was made in modelling clay by rolling the clay into a ball and pressing the targeted finger into the clay for the S7 edge. The mold was then put in the refrigerator so it would better hold its shape. For the S10, the mold was made of candle wax, due to it being performed on another time where the modelling clay weren't available.

We would have liked to do the attacks truly from the ground up, with dusting for fingerprints off of the phone, making a mold and fake fingerprint from the dusted print, but we found that this would demand a lot more equipment than we had access to at the time of performing the attacks.

To make the fake print, we bought gelatin from the store, and mixed a thick mixture of gelatin and water. The gelatin was heated in a water bath and stirred until all the gelatin had mixed with the water. Then, we let the mixture cool, before heating it up again. This was done until all the bubbles in the gelatin was gone, and it had a thick and rubbery consistency. When the consistency of the gelatin was right, we poured the gelatin into the mold and put it into the freezer.

Since the process takes a while and we made more gelatin than we needed for one mold, we made another mold for another finger that is also a finger that should be authenticated by the phone. This was done for the attack on the S7 edge.

The sensor on most phones today is capacitive. A capacitive sensor works because of the conductivity of the human skin. When the finger makes contact with the sensor, the skin and sensor surface form a capacitor. Gelatin should work on capacitive sensors, because of its water content. The S10 though, uses a new sonic fingerprint sensor that is located underneath the display of the phone. The fingerprint sensor works by sending sonic waves through the display and reading how these waves bounce back off of the finger that is laid over the sensor. This allows the sensor to create a 3D reproduction of the finger on the sensor, which should be more secure and harder to forge than capacitive and optical sensors that only reproduce 2D images. Since our attack involves us having a good mold of the real fingerprint, we might be able to fool the sonic sensor as well.

**Galaxy S7 edge**

The first try with the first mold did not let us into the phone. The phone did register that there was a "finger" on the sensor, but did not find a match in its database. Upon inspection of the gelatin

fingerprint, the mold might be too deep, thus making too little of the gelatin fingerprint covers the sensor for it to be admitted. In the second try, we will try to make our mold flatter, so more of the gelatin fingerprint will cover the sensor.

The first try with the second mold did let us into the phone after a lot of tries and different positioning of the finger. The first time we got in was after about 10 minutes of constantly trying to get into the phone. When doing this we unlocked the phone with the help of the PIN code when the fingerprint sensor "shut down" after to many unsuccessful tries so we did not have any downtime. This means that having a prior estimation of where the victim places their finger on the sensor would be beneficial for the attacker, saving some time for them.

**Galaxy S10**

The Galaxy S10 also offers fingerprint authentication with a in-screen sonic fingerprint sensor, which builds a 3D image of the user's fingerprint. We deployed the same attack against the this phone as the Galaxy S7 edge. We also used the same recipe and the same mold. With the mold we made, we had no success in getting into the phone. The amount of testing was on par with that of the Galaxy S7 edge, indicating that this sensor is more resilient against this type of attack than the Galaxy S7 edge's sensor and possibly other capacitive sensors. A short time after we concluded our attack against the Galaxy S10, a guy with the username darkshark posted a video on Imgur showing him unlocking his Galaxy S10 with a 3D printed fingerprint he got from a photo he took with the same phone, as we described in section 2, showing it is possible to also fool the Samsung Galaxy S10 fingerprint sensor.

### 3.1.2 Facial recognition attack

For the facial recognition attacks, we conducted attacks against both a Galaxy S10 and an iPhone XR. The Galaxy phone uses 2D facial recognition while the iPhone uses 3D facial recognition. 3D recognition is shown to be way more secure and much harder to fool by holding a photo or video in front of the sensor.

**Galaxy S10**

When we conducted the attack on Samsung's new flagship phone, we first set the facial recognition up using the main authors face. We unsuccessfully tried to use an image taken on the same day as the facial recognition authentication was set up on the phone, showing the image on another phone, a 15 inch laptop and a 24 inch display. We also made an unsuccessful attempt on the same displays using a video instead of an image. We made multiple attempts on all the different displays, with varying the distance between the phone and the image/video. We also tried going on the internet trying to find images of the test person and trying to authenticate them that way, imitating a more real life attack, but were also unsuccessful with these attempts. Even though we deem these instances of the attack to be unsuccessful, we did manage to get authenticated twice with the above methods, but the amount of time and tries we had to make for these two successful authentications to be made, the chance for an attacker to unlock the phone would be more luck based than knowing what to do.

Before we decided to give up on the attack completely, we had the girlfriend of the main author

register her face on the phone. We then took a picture of her with another phone, and showed it to the S10, which unlocked the phone immediately. We then tried to turn off the faster unlock option that sacrifices security for speed, and tried the same thing again. The phone still unlocked rather quickly, making the attack very successful.

Taking the attack a step further, we acquired an image from the internet of the test person, but we were not able to get authenticated by the S10 with this approach. This might be due to there not being any recent images of good enough "attack quality" of the test person on any major social media. With attack quality we mean any image where the test person have the correct head position so the whole face can be seen, correct head tilt, etc.

**iPhone XR**

As with the attacks on the two different fingerprint sensors, we conducted the same facial recognition attacks on the iPhone XR as we did on the Galaxy S10. Unsurprisingly, the attacks did not work against the iPhone XR, as it uses 3D recognition, and all our attacks was mostly directed at fooling 2d sensors. Though we did not manage to fool the iPhone's facial recognition, it has been done by security experts at Bkav [41].

## 3.2 Questionnaire implementation

In this section we will describe why and how we did our questionnaire. The full questionnaire is enclosed in the appendix.

Since user-friendliness of different authentication methods is a highly subjective matter we decided to make a questionnaire. Our questionnaire was made with the intent to collect data about the user-friendliness of different authentication methods on mobile phones. The questionnaire is short, and takes about 2-3 minutes to answer. This is a positive feature of the questionnaire, as more people will want to answer when it doesn't take a lot of their time. The questionnaire was distributed on Facebook, where friends and family helped share the questionnaire so it would reach more people. This yielded good responses with a little over 130 people answering the questionnaire. The questionnaire was also distributed by e-mail inside Statens kartverk. This provided more people of different age groups to answer, as a lot of people that answered from Facebook probably were under 30 years old. From Kartverket, we estimate that we got around 120 answers. Our total number of responses were 261, but some of these might need to be filtered out if the questionnaire isn't finished and therefore cannot give us meaningful data.

### 3.2.1 Questionnaire description

The questionnaire employs condition logic to the questions, so respondents don't have to answer or view questions that are irrelevant to them. If a respondent answers that he is using fingerprint along with PIN authentication, the respondent will only receive questions about those two authentication methods. The questionnaire starts by asking the respondent about their age, gender, if they use a phone with IOS, Android, or any other operating system and if they use a lock screen for their device. If their answer is not to this question, they get asked why, and the questionnaire ends. If they answer that they do use a lock screen to protect their device, the respondent moves on to be

asked what they are using or have used before. After this, most of the questions are similar to each other, with two "groups" of questions, one for secret based authentication, and one for biometric authentication. For both PIN and password, the respondent is asked how many characters are in their PIN/password along with how user-friendly they think each authentication method is. For biometric authentication, how many times the biometric authentication method is deactivated because of to many false rejects in a row, where the user now has to enter their PIN instead to authenticate themselves is asked, along with how user-friendly the respondent think the authentication method is.

The questionnaire is anonymous, and there is no way of tying a specific respondents answers to a specific person based purely on the data we collected in our questionnaire. Age and gender is not enough to discern someones identity, so the questionnaire is anonymous. This is important since we are asking people to give us information about their authentication methods, like how many characters that are in their PINs or passwords. This is information that could help malicious actors in breaking in to a targets mobile phone, so it is important that the information we collect and include in our thesis is anonymous.

## 3.3 Data analysis and results

After giving out the questionnaire, we waited for about two months to get replies. After these two months, 261 people had at least looked at our questionnaire. Of these 261, 253 of them had completed the questionnaire. This should be enough respondents to give us some data about how user friendly different authentication methods are.

## 3.4 General information

Of the 253 respondents, 149 of them are female and 104 for of them are male. The age variation is quite evenly distributed between all age groups except 65+. This can be seen in figures 8 and 9.

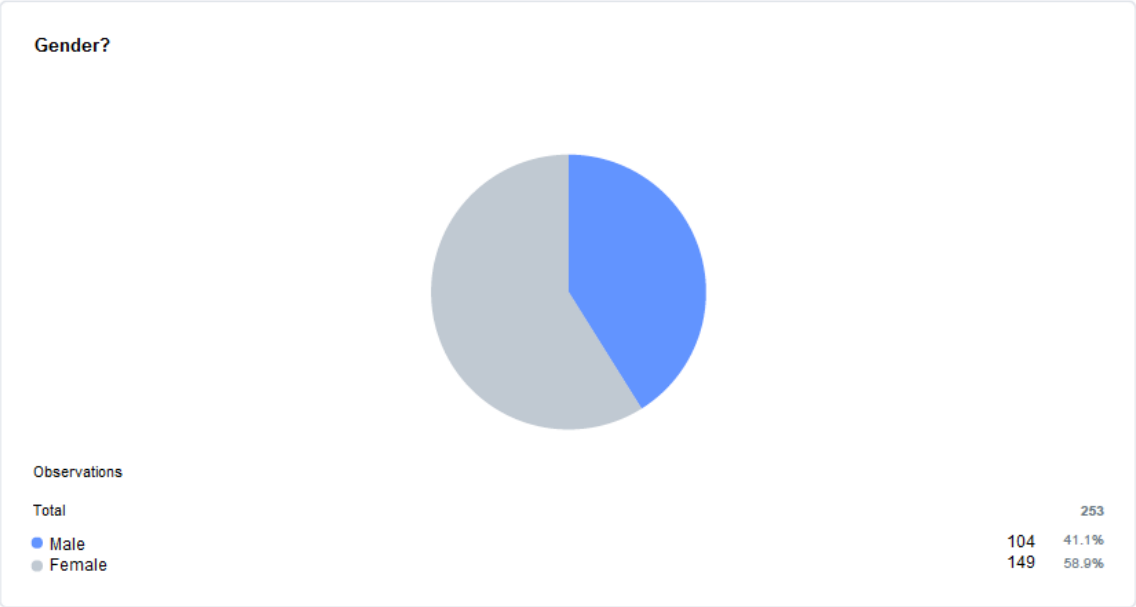


Figure 8: Gender distribution of questionnaire respondents

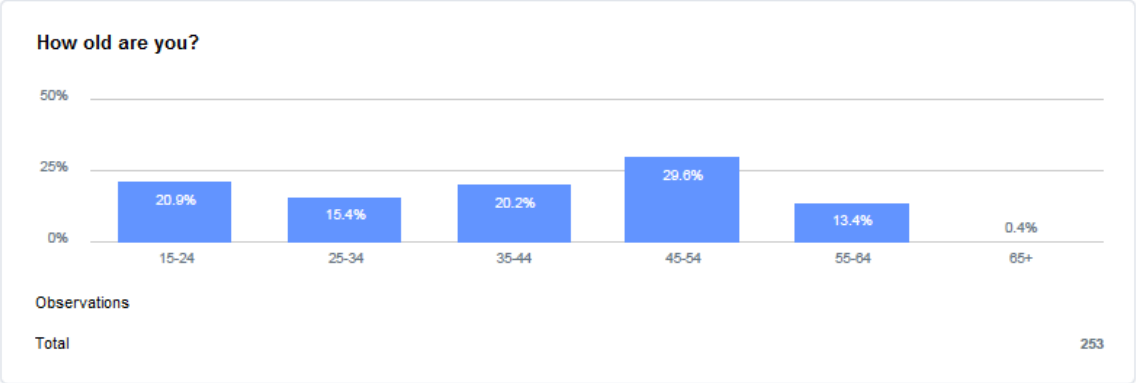


Figure 9: Age distribution of questionnaire respondents

In figure 10 we also see that there are more people with iPhones than Android phones that answered the questionnaire, showing 144 people saying they are using an iPhone, 106 people said they are using an Android, and 3 people said they were using another type of phone/operating system. Upon further inspection of the data, all of the three people that had answered "other" did in fact use an android phone. The reason they answered "other" could be the strong association

between Samsung and Android for people that are not very interested in technology or mobile phones, and maybe thinking that "my phone is not a Samsung phone, so it is not an Android."

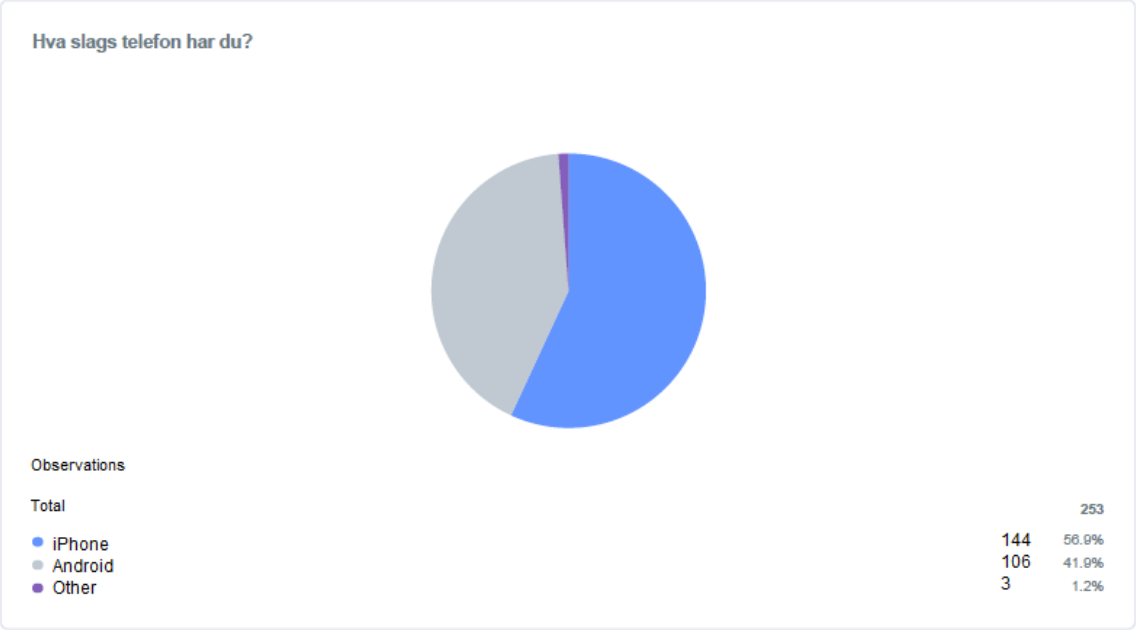


Figure 10: Phone distribution of questionnaire respondents

## 3.5 Use of lock screen

As shown in figure 11, on the question about the use of lock screen, 247 respondents answered that they are using a lock screen for their phone, while 6 answered that they are not using a lock screen.

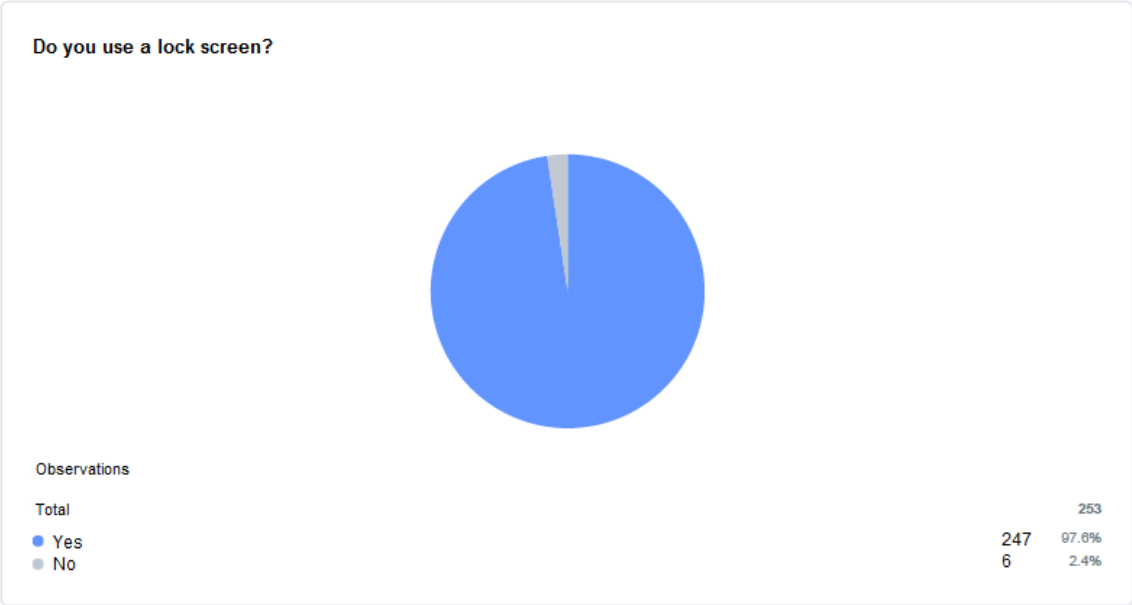


Figure 11: Use of a lock screen amongst respondents

As a follow up question for those that answered they are not using a lock screen, we gave them a question asking them to state why they have made this choice. The five respondents that answered this question stated (translated from Norwegian):

- Respondent 1 don't know - a little inconvenient
- Respondent 2 I think it is inconvenient
- Respondent 3 I am using facial recognition.
- Respondent 4 Time
- Respondent 5 Never had my phone stolen, and I have nothing to hide from those who are allowed to use my phone.

It is clear that one of the respondents actually use a lock screen for their phone, but did not understand that using facial recognition to unlock your phone is using a lock screen.

Of the 5 respondents that correctly answered that they are not using a lock screen, one belonged to the age group 15-24, two belonged to the 25-34 age group, one belonged to the age group 45-54 and one belonged to the 55-64 age group. The fact that 3 of 5 were in the age group 15-34 was a little surprising, as we tend to think that younger people worry and care more about securing their phone and data.

## 3.6 Use of authentication methods

When it comes to what the respondents used or had used before, PIN and fingerprint recognition is the two most popular authentication methods by far, with respectively 83.4 and 70.4 percent of respondents had used or are using these two authentication methods. Use of a password is also fairly popular, with 29.4% of respondents having tried using that as an authentication method. Pattern authentication is also well represented with 23.4%. The second most used biometric authentication method is facial recognition, with 13.7% of respondents saying they are using or have used it. Iris recognition usage comes in at 3.6%, while the "other" category contained two answers. One respondent had tried the LG knock code, which is basically a less secure version of a normal PIN code. The LG knock code is a 2x2 grid where the user can enter 3-8 "knocks". The positive side of the LG knock code is that it can be entered from standby, letting the user skip having to turn the screen on before authenticating themselves. The other respondent answered that he used multi-factor authentication, but did not say which authentication methods.

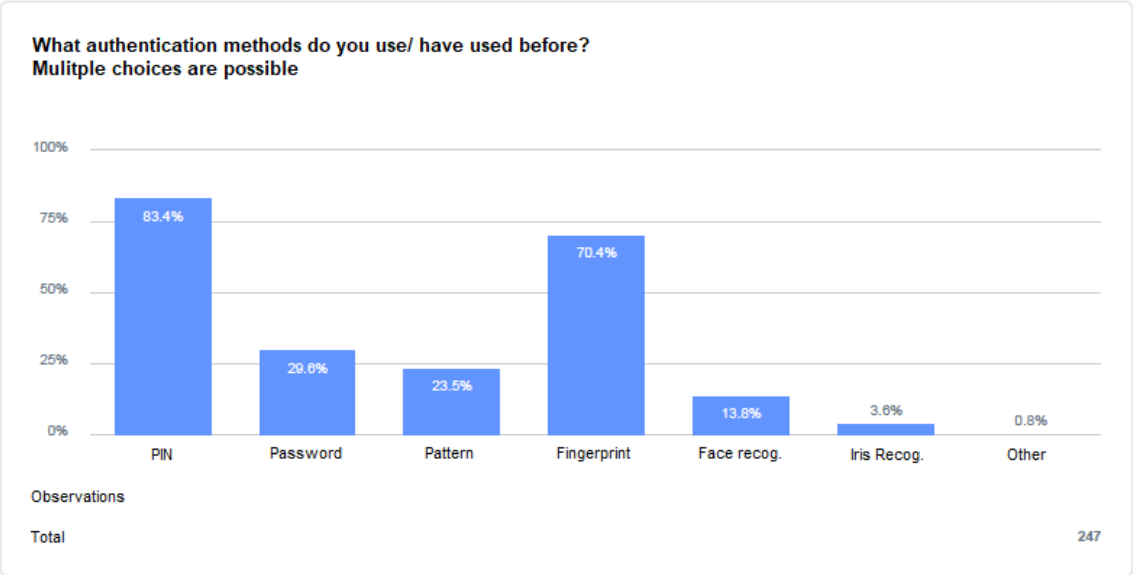


Figure 12: Percentage of respondents that have tried different the different authentication methods

### 3.6.1 PIN code

As showed above, the use of PIN code is the most widespread authentication mechanism for use on mobile phones. This is to be expected, as biometric authentication methods require the user to also have a PIN or a password in case the biometric authentication fails. Most of the respondents say that their PIN code is 4 or 6 digits long, and together these two PIN lengths make up 96.1% of the answers. None of the respondents used a PIN code that were longer than 8 digits. In figure 13,

all the answers are shown. Since no one answered that their PIN code was more than 8 digits long, those alternative answers is not pictured in this section.

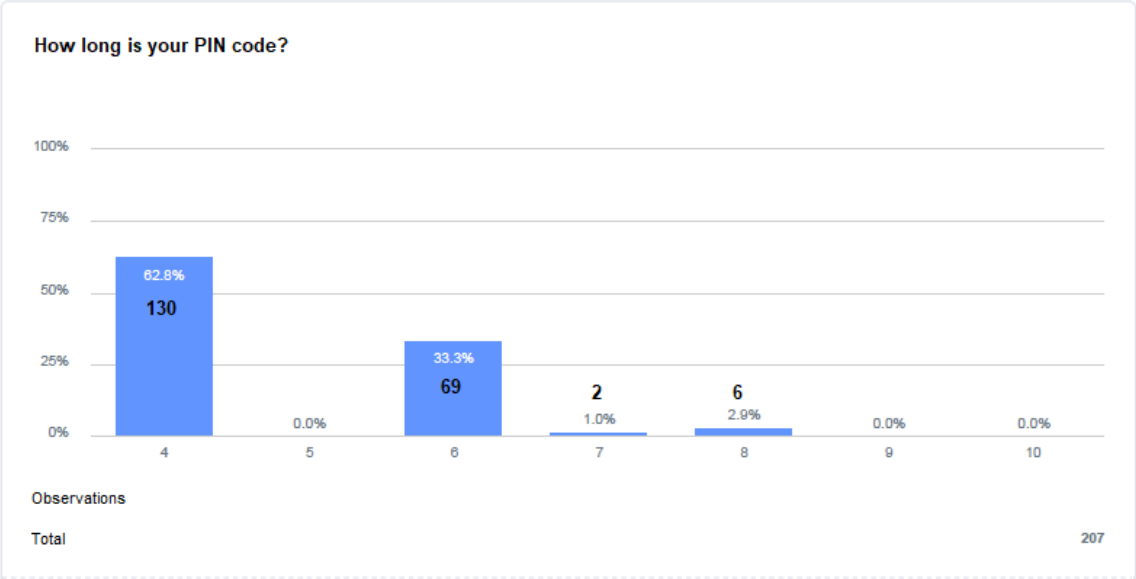


Figure 13: Percentage and number of respondents using different length PIN codes.

The average length of a PIN for our respondents is 4.82 digits, and the standard deviation is low, at 1.111.

Most of our respondents use PIN authentication, so the user-friendliness score we find for this category will be more representative than some of the other authentication methods featured in our questionnaire. Figure 14 shows that most of PIN authentication users think that it is on the top half of the user-friendliness scale. The mean user-friendliness score reflects this, coming out at 4.33 and a standard deviation of 1.283.

**PIN - user friendliness**

| | Frequency | Valid Percent |
|---|---|---|
| 1 | 5 | 2,4 |
| 2 | 13 | 6,3 |
| 3 | 32 | 15,5 |
| 4 | 60 | 29,1 |
| 5 | 51 | 24,8 |
| 6 | 45 | 21,8 |
| Total | 206 | 100,0 |

Figure 14: Frequency of PIN user-friendliness scores.

### 3.6.2 Password

Of the 73 respondents that uses or have tried password authentication 54.8% of them use a complex password that includes at least one letter, one capital letter, one number and one special character.


Figure 15: Use of complex passwords

In figure 16 the length of the respondents passwords is shown to range from 4 characters to 16+ characters. Most respondents use a password with the length of 4, 6 or 8 characters, with each

of the categories consisting of 18 to 22 percent of the answers. The average length of a password for our respondents is 7.90 characters long, with a standard deviation of 3.202 characters. In this calculation we have assumed that the people who responded that their password is 16+ characters long have passwords that are exactly 16 characters long, as we don't know the exact number of characters their passwords contain.

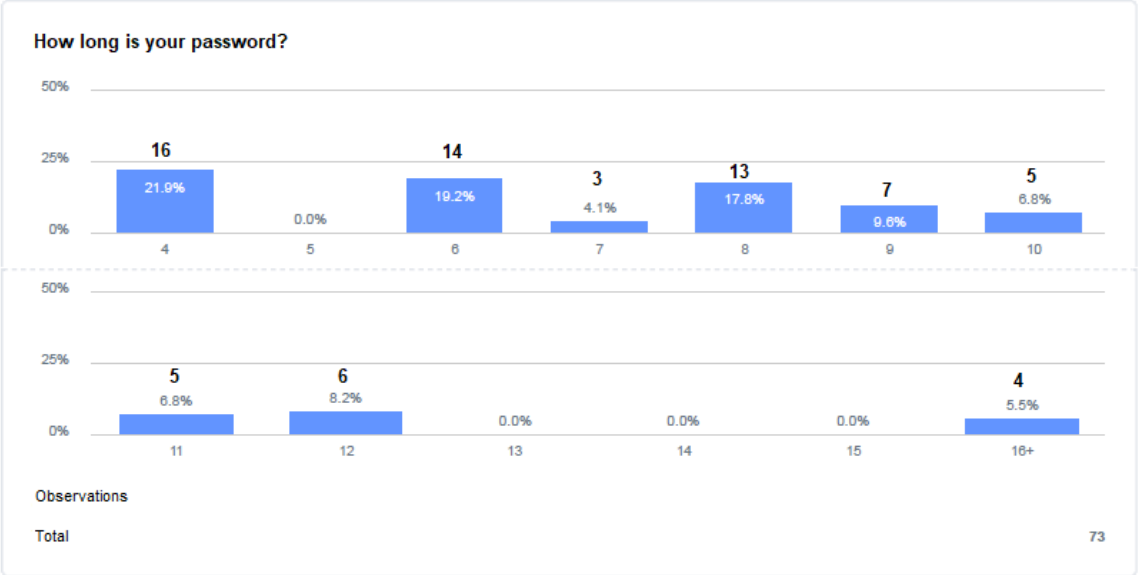


Figure 16: Length of respondents passwords

Shown in figure 17, we see that the user-friendliness scores are spread out across the entire scale. Surprisingly, a lot of our respondents think that password authentication is a 5 or 6 on the user-friendliness scale. This is surprising because we thought that password authentication would be viewed as non-friendly because of the effort that needs to be undertaken to enter them on a small screen.

**Password - user friendliness**

| | Frequency | Valid Percent |
|---|---|---|
| 1 | 5 | 6,8 |
| 2 | 5 | 6,8 |
| 3 | 15 | 20,5 |
| 4 | 21 | 28,8 |
| 5 | 15 | 20,5 |
| 6 | 12 | 16,4 |
| Total | 73 | 100,0 |

Figure 17: Frequency of password user-friendliness scores.

Even though more people than expected gave a high user-friendliness score, the mean score of password authentication is not more than 3.99 with relatively high standard deviation of 1.409.

### 3.6.3 Pattern

For the pattern authentication method we did not ask respondents any other questions than how user-friendly they felt the method is. We can see from the graphic that most of the respondents think that pattern authentication is user-friendly, with most of the answers being from a 4 to a 6. This is shown in figure 18.

**Pattern - user friendliness**

| | Frequency | Valid Percent |
|---|---|---|
| 1 | 1 | 1,7 |
| 2 | 5 | 8,6 |
| 3 | 7 | 12,1 |
| 4 | 14 | 24,1 |
| 5 | 16 | 27,6 |
| 6 | 15 | 25,9 |
| Total | 58 | 100,0 |

Figure 18: Frequency of pattern user-friendliness scores.

From the data, we have calculated the mean user-friendliness score to be 4.45, with a standard deviation of 1.327. We expected pattern authentication to be well liked by users, since it provides a non-biometric fast and easy way of unlocking a phone.

### 3.6.4 Fingerprint

Fingerprint authentication was the second most used authentication method after PIN-codes. It is clear that there is a wide spectrum on how often fingerprint authentication fails enough times for it to be deactivated and prompting the user to enter their PIN code. We also see that there is some differences between the respondents who uses an iPhone and those who use Android phones. This difference is shown in figure 19, where we can see that iPhone users tend to get locked out of their phone more often than those who use an Android phone. The difference is as large as 11.3% and 10% for the weekly and daily categories respectively.

**Fingerprint - iPhone**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | Never | 17 | 15,0 |
| | Rarely | 25 | 22,1 |
| | Monthly | 18 | 15,9 |
| | Weekly | 35 | 31,0 |
| | Daily | 15 | 13,3 |
| | Multiple times daily | 3 | 2,7 |
| | Total | 113 | 100,0 |

**Fingerprint - Android**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | Never | 14 | 23,0 |
| | Rarely | 19 | 31,1 |
| | Monthly | 13 | 21,3 |
| | Weekly | 12 | 19,7 |
| | Daily | 2 | 3,3 |
| | Multiple times daily | 1 | 1,6 |
| | Total | 61 | 100,0 |

Figure 19: Comparison of how often a passcode (PIN, password, pattern) is required because of too many unsuccessful fingerprint match attempts for both iPhones and Android.

In figure 20 we show the frequencies of user-friendliness rating for both iPhone and Android users. Figure 21 shows the mean user-friendliness score for iPhone and Android users along with the mean score for all answers. As seen, Android users think that fingerprint authentication is more user-friendly than iPhone users do, with Android fingerprint authentication being reported as 0.13 points ahead of iPhone fingerprint authentication. The standard deviation for Android users is also lower than it is for iPhone users.

**iPhone - User-friendliness fingerprint**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | 1 | 3 | 2,7 |
| | 2 | 7 | 6,3 |
| | 3 | 11 | 9,8 |
| | 4 | 22 | 19,6 |
| | 5 | 35 | 31,3 |
| | 6 | 34 | 30,4 |
| | Total | 112 | 100,0 |

**Android - User-friendliness fingerprint**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | 2 | 5 | 8,2 |
| | 3 | 5 | 8,2 |
| | 4 | 12 | 19,7 |
| | 5 | 17 | 27,9 |
| | 6 | 22 | 36,1 |
| | Total | 61 | 100,0 |

Figure 20: Comparison between iPhone and Android users on how user-friendly fingerprint authentication is.

| | N | Minimum | Maximum | Mean | Std. Deviation |
|---|---|---|---|---|---|
| iPhone - Fingerprint | 112 | 1 | 6 | 4,62 | 1,324 |
| Android - Fingerprint | 61 | 2 | 6 | 4,75 | 1,260 |
| All - Fingerprint | 173 | 1 | 6 | 4,66 | 1,300 |

Figure 21: Comparison of mean user-friendliness scores of fingerprint authentication.

From figure 22, we see that there is that there is some correlation between how often users fingerprint authentication gets deactivated due to too many false rejections and how user-friendly users think that fingerprint authentication is. A negative relation, as we have in our case, means that our variables are moving in opposite directions. The values of how fingerprint authentication gets deactivated has been converted to numbers in our analysis tool, where the "Never" option is 1 and "Multiple times daily" gets the value 6. This is why the correlation is a negative value, since users who never gets locked would be more likely to give the authentication method a higher user-friendliness score.

**Correlation iPhone**

| | | @15 | @19 |
|---|---|---|---|
| Fingerprint - deactivated | Pearson Correlation | 1 | -,443** |
| | Sig. (2-tailed) | | ,000001 |
| | N | 113 | 112 |
| Fingerprint - user friendliness | Pearson Correlation | -,443** | 1 |
| | Sig. (2-tailed) | ,000001 | |
| | N | 112 | 112 |

**. Correlation is significant at the 0.01 level (2-tailed).

**Correlation Android**

| | | @15 | @19 |
|---|---|---|---|
| Fingerprint - deactivated | Pearson Correlation | 1 | -,439** |
| | Sig. (2-tailed) | | ,0004 |
| | N | 61 | 61 |
| Fingerprint - user friendliness | Pearson Correlation | -,439** | 1 |
| | Sig. (2-tailed) | ,0004 | |
| | N | 61 | 61 |

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 22: Correlation between the amount of times a respondent gets prompted for a passcode because of too many unsuccessful attempts and the user-friendliness score.

### 3.6.5 Facial recognition

Facial recognition coming in with such a low amount of usage compared to fingerprint recognition does not come as a surprise. Older phones did not offer facial recognition with good security or offer it at all, so facial recognition was largely popularised with the launch of Apple's FaceID with their iPhone X. The total number of responses for facial recognition is 34, where 19 of those are from iPhone users and 155 are from Android users. Figure 23 shows that facial recognition never of rarely locks out over half of the users, both for iPhone and Android phones. After that, most iPhone users report that they get locked out on a weekly basis, while Android users reports were more evenly distributed.

**Facial recognition - iPhone**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | Never | 5 | 26,3 |
| | Rarely | 5 | 26,3 |
| | Monthly | 1 | 5,3 |
| | Weekly | 7 | 36,8 |
| | Daily | 1 | 5,3 |
| | Total | 19 | 100,0 |

**Facial recognition - Android**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | Never | 1 | 6,7 |
| | Rarely | 7 | 46,7 |
| | Monthly | 2 | 13,3 |
| | Weekly | 2 | 13,3 |
| | Daily | 3 | 20,0 |
| | Total | 15 | 100,0 |

Figure 23: Comparison of how often a passcode (PIN, password, pattern) is required because of too many unsuccessful face match attempts for both iPhones and Android.

Figure 24 shows the frequency of the user-friendliness scores for facial recognition. Almost all of the answers for iPhones gives facial recognition a score of 5 or 6. For Android users, the answers are more evenly divided between scores, with most respondents giving it a user-friendliness score

of 4. Figure 25 shows how big the difference in mean user-friendliness scores, with an iPhone user mean of 5.21 with a standard deviation of 0.855. This is the highest mean and lowest standard deviation we have recorded. User-friendliness of Android facial recognition on the other hand only has a mean score of 4.20, a whole point below the iPhone counterpart. The standard deviation is also a lot higher, at 1.32. Since there are only 19 and 15 user responses in each of the groups, our results might not be completely accurate, but they give an idea of what is thought about the authentication method. Also, since the user-friendliness score is so different between iPhones and Android phones, we will be using the two different scores in the function. The iPhone mean score will be used for 3D facial recognition, while the Android mean score will be used for 2D facial recognition.

**iPhone - User-friendliness facial recognition**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | 3 | 1 | 5,3 |
| | 4 | 2 | 10,5 |
| | 5 | 8 | 42,1 |
| | 6 | 8 | 42,1 |
| | Total | 19 | 100,0 |

**Android - User-friendliness facial recognition**

| | | Frequency | Valid Percent |
|---|---|---|---|
| Valid | 2 | 2 | 13,3 |
| | 3 | 2 | 13,3 |
| | 4 | 5 | 33,3 |
| | 5 | 3 | 20,0 |
| | 6 | 3 | 20,0 |
| | Total | 15 | 100,0 |

Figure 24: Comparison between iPhone and Android users on how user-friendly facial recognition is.

| | N | Minimum | Maximum | Mean | Std. Deviation |
|---|---|---|---|---|---|
| iPhone - Facial recognition | 19 | 3 | 6 | 5,21 | ,855 |
| Android - Facial recognition | 15 | 2 | 6 | 4,20 | 1,320 |
| All - Facial recognition | 34 | 2 | 6 | 4,76 | 1,182 |

Figure 25: Comparison of mean user-friendliness scores of facial recognition.

From figure 26, we see that there is a moderate negative correlation between how often a passcode is required because of too many failed authentication attempts in a row and how user-friendly

facial recognition is reported to be. For Android phones, there is a strong negative correlation, with a value of -0.843.

**Correlations iPhone**

| | | deactivated | user friendliness |
|---|---|---|---|
| Face rec - deactivated | Pearson Correlation | 1 | -,507* |
| | Sig. (2-tailed) | | ,027 |
| | N | 19 | 19 |
| Face rec - user friendliness | Pearson Correlation | -,507* | 1 |
| | Sig. (2-tailed) | ,027 | |
| | N | 19 | 19 |

*. Correlation is significant at the 0.05 level (2-tailed).

**Correlations Android**

| | | deactivated | user friendliness |
|---|---|---|---|
| Face rec - deactivated | Pearson Correlation | 1 | -,843** |
| | Sig. (2-tailed) | | ,00008 |
| | N | 15 | 15 |
| Face rec - user friendliness | Pearson Correlation | -,843** | 1 |
| | Sig. (2-tailed) | ,00008 | |
| | N | 15 | 15 |

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 26: Correlation between the amount of times a respondent gets prompted for a passcode because of too many unsuccessful attempts and the user-friendliness score.

### 3.6.6 Iris recognition

There are not a lot of phones that offer iris recognition, with the last couple of generations of Samsung Galaxy S-series and Note series being the most popular [66], of those phones. On their newest phones, Samsung have decided to leave out the iris scanner, leaving even fewer phones with iris authentication available. Since there is no iPhone with iris recognition on the market today, there won't be any comparisons between iPhones and Android in this subsection. There is also only 9 people in total that responded that they use or had used iris recognition, so the result from this analysis is probably not 100% precise. Nonetheless, we will use the results of the analysis in the sorting function. As seen in figure 27, 6 out of 9 respondents get locked out of their phone using iris recognition on a weekly or daily basis, while 3 responded that they are rarely locked out.

**Iris**

| | Frequency | Valid Percent |
|---|---|---|
| Rarely | 3 | 33,3 |
| Weekly | 4 | 44,4 |
| Daily | 2 | 22,2 |
| Total | 9 | 100,0 |

Figure 27: How often a passcode (PIN, password, pattern) is required because of too many unsuccessful iris match attempts.

In figure 28 it is shown that most of the user-friendliness scores hover on the lower end of the scale. None of our respondents gave the authentication method a score of 1 or 6, but this

is most likely due to the low number of respondents for this authentication method. The mean user-friendliness score for iris recognition is the lowest we recorded, at only 3.22, with a standard deviation of 1.093. Again, with a sample size of only 9, the results might not be very precise, but it gives us an idea of how users tend to think about the current implementations of iris recognition on mobile phones.

**Iris**

| | Frequency | Valid Percent |
|---|---|---|
| 2 | 3 | 33,3 |
| 3 | 2 | 22,2 |
| 4 | 3 | 33,3 |
| 5 | 1 | 11,1 |
| Total | 9 | 100,0 |

Figure 28: Frequency of reported user-friendliness score.

In figure 29 we see that unlike the other biometric authentication methods, there is no real correlation between how often a passcode (PIN, password, pattern) is required because of too many unsuccessful match attempts in a row and user-friendliness score, especially since the correlation value that we got from this authentication method is positive, not negative as we would expect. This might be due to the very low amount of answers we got for the use of iris recognition, and that with more answers, a correlation might have been revealed.

**Correlations - Iris**

| | | deactivated | user friendliness |
|---|---|---|---|
| Iris - deactivated | Pearson Correlation | 1 | ,082 |
| | Sig. (2-tailed) | | ,833 |
| | N | 9 | 9 |
| Iris - user friendliness | Pearson Correlation | ,082 | 1 |
| | Sig. (2-tailed) | ,833 | |
| | N | 9 | 9 |

Figure 29: Correlation between the amount of times a respondent gets prompted for a passcode because of too many unsuccessful attempts and the user-friendliness score.

### 3.6.7 User-friendliness recap

For the user-friendliness scores we used the average of the respondents answers for each category. Though the scale we gave the respondents for rating the user-friendliness was only from 1-6 the mean scores are much closer than we thought they would be, with the least and most user-friendly authentication method only being about 1.5 points apart. Since the different authentication methods have a different amount of answers collected, how certain we can be that a score represents the bigger population varies. Even though this is the case for some of the authentication methods, we will use the scores we have got from the respondents in our questionnaire when using our function for sorting the best authentication method, regardless of the number of people who responded.

| | Authentication method | User-friendliness score (mean) | Std. deviation | # of responses |
|---|---|---|---|---|
| 1 | Facial recognition | 4.76 | 1.182 | 206 |
| 2 | Fingerprint | 4.66 | 1.300 | 73 |
| 3 | Pattern | 4.45 | 1.327 | 58 |
| 4 | PIN | 4.33 | 1.283 | 174 |
| 5 | Password | 3.99 | 1.409 | 34 |
| 6 | Iris recognition | 3.22 | 1.093 | 9 |

Table 9: Sorted list of authentication method scores

We also see that even though the scores are close, the authentication methods have been placed more or less in the order we imagined they were going to be placed in. On the top of the list we have facial recognition, followed closely by fingerprint recognition. As stated above, the amount of people responding to having used facial recognition was very low compared to the ones that said they had tried fingerprint recognition, so the mean score of facial recognition could have changed if more of the respondents had tried the authentication method, but at the same time, the standard deviation for the method is small, indicating that there were less variance for facial recognition than in most of the other authentication methods. As use of fingerprint recognition had a lot of answers, we can be fairly certain that the result of the mean score and standard deviation is representative. The standard deviation is higher for fingerprint recognition than for facial recognition, showing there are more variance in how user-friendly users think fingerprint recognition is. Below fingerprint, we find the first non-biometric authentication method. Pattern authentication scores 0.21 points behind fingerprint, with a slightly higher standard deviation. PIN authentication was the most used authentication method by our respondents, so we can be fairly certain that the scores we got from PIN authentication is representative for the population. The score for PIN is 4.33 with a standard deviation of 1.283. Password authentication is the first authentication method to drop under a score of 4, and comes in at 3.99 with a standard deviation of 1.409. This is the highest standard deviation out of all the authentication methods, showing that respondents password authentication is an authentication method were peoples opinions varies more than other authentication methods. The least user-friendly authentication method according to our respondents is iris recognition, with a score of 3.22, clearly lower than even password authentication. This might have something to

do with there only being 9 people who responded on the use of iris recognition, but the score is indicative that iris recognition is not a very user-friendly authentication method on mobile phones today. The popularity of iris recognition seem to be generally low, as Samsung has removed their iris recognition feature on their latest generation devices. Due to this, we don't feel the need to conduct more in-depth interviews with the people that had used iris recognition.

## 3.7 Function

When making the function that calculates and sorts the best authentication methods, we chose to use Python. The function takes two inputs which are used as scales so the user can decide how much both security and user-friendliness matter to them. The information of the different authentication methods are collected from a CSV file (enclosed in the appendix). This file can easily be edited to add, remove or alter the information on authentication methods. Currently, the function supports authentication methods of both the "secret" and "biometric" categories of authentication, where the "secret" category's security is measured in bits of entropy, and the "biometric" category's security is measured in the false accept rate of the authentication method. The function converts these values of either entropy or FAR to a scale of 1-6 so the security values are on the same scale as the user-friendliness values. After the combined score of security and user-friendliness is calculated, they are sorted and and printed to screen and written to file for later use. Every time the function is run, the file is overwritten so that it doesn't get too large and hard to read.

Listing 3.1: Ranking function

```
def makelist(security = 10, friendliness = 10):
        import csv
        import math
        from operator import itemgetter

        #Blanks out the output text file
        f = open('E:\Master\Masteroppgave\\output.txt', 'w')
        f.write('')
        f.close()

        lis = []

        #Opens and read the CSV file
        with open('E:\Master\Masteroppgave\\authentication_methods.csv') as file:
                reader = csv.DictReader(file)

                #Computes the combined score of the authentication methods row for row
                for row in reader:
                        if row['type'] == 'secret':
                                if float(row['security']) > 65:
                                        row['security'] = 65
                                numeric_score = (((float(row['security']) - 7) * 5) / (65
                                     - 7)) + 1

                        if row['type'] == 'biometric':
                                if float(row['security']) > 0.015:
                                        row['security'] = 0.015
                                numeric_score = 6 - (5/0.015) * math.sqrt(0.015 * float(
                                    row['security']))

                        combined_score = (numeric_score * float(security)) * (float(row['
                            friendliness']) * float(friendliness))
                        row["combinedscore"] = int(combined_score)
                        lis.append(row)


        #Sorts and prints the authentication methods
```

```
        print('\r')
        sortedlis = sorted(lis , key=itemgetter('combinedscore'),reverse = True)
        for x in sortedlis:
                f = open('E:\Master\Masteroppgave\\output.txt', 'a')
                f.write(str(x) + '\r')
                f.close()
                print(x)


#Presents the user with question of how they want to weigh security and user-friendliness
     and runs the function based on this input
sec = int(input('How important is security on a scale from 1-10?: '))
friend = int(input('How important is user-friendliness on a scale from 1-10?: '))
while sec < 1 or sec > 10:
        sec = int(input('The number you input for security is not between 1 and 10, 
            please input such a number: '))
while friend < 1 or friend > 10:
        friend = int(input('The number you input for user-friendliness is not between 1 
            and 10, please input such a number: '))

makelist(sec, friend)
```

# 4 Discussion

This chapter is a discussion of the results presented in section 3. The first three sections discuss the research questions from section 1, while the other sections discuss the different authentication methods. The last section discusses the limitations of this research.

## 4.1 The best authentication method based on security and user-friendliness

When we look at both the security and the user-friendliness of the different authentication methods, we see that 3D facial recognition emerges at the top in the biometric category, as well as being the most user-friendly authentication method. Both face and fingerprint recognition score high in user-friendliness, while iris recognition is the least user-friendly authentication method of all our proposed methods. The user-friendliness of the non-biometric authentication methods seems to be ranked based on how easy each authentication method is supposed to be in use, with pattern being easier and faster than PIN, and PIN being easier and faster than password.

## 4.2 Measuring the security of different solutions

Trying to measure the security of authentication methods across different categories (secrets, biometric) proved to be a difficult task, especially because of the focus on mobile phones and what comes with that. There exists some research trying to convert the randomness and uncertainty in biometric authentication methods to entropy [67, 68], but this proved difficult, as it required us to get biometric data we did not have access to. For fingerprints, the sensor is also a lot smaller than normal, so taking the converted entropy values in the research articles would not be correct either. We ended up sticking with the two different measures of security, entropy for PINs, passwords and patterns, and false accept rate for fingerprint, face and iris recognition. We found no good way of measuring the security of a biometric system under attack. The solution to this problem for us was to note this in the output of the system. When ranking the authentication methods, the two different measures of security is converted to the same 1-6 scale as we measured user-friendliness, based on some criteria. This will of course be somewhat of an estimation, but if we decided to, and could convert to entropy instead, the entropy value would also be an estimation.

## 4.3 Considering user-friendliness

The user-friendliness of an authentication method is dependent on the ease of use, intuitiveness, speed and reliability. If an authentication method is slow and requires a lot of work to use, or the reliability is low where the authentication method stops working, the user-friendliness of that authentication method will be lower than that of an authentication method that is doing all those factors better. When we calculated the user-friendliness score for the different authentication meth-

ods, we did expect to see the biometric authentication methods on top. These methods require less work, are generally faster than typing or swiping, and although the user might need to place their finger on the fingerprint sensor more than one time on some occasions, users seems to still find it better for them to use. This is also most likely why the iris recognition method is given the lowest score of all, as the Samsung phones with this authentication method available seems to have had some problems with the scanner stopping to work partially or completely until manual action to fix the problem like rebooting the phone, clearing iris scanner cache, updating every app, deleting saved biometric data and setting up again and more. [69]. This leads to iris recognition being hard to use over time, as well as having low reliability.

## 4.4 Choices of PINs, passwords and patterns

### 4.4.1 PIN

The choices in length of PIN clearly shows that for most users, the PIN is not used to be secure, but rather convenient. Most users have a PIN code that is 4 digits long, which provides 10,000 different combinations, but as seen in section 2, the distribution of PINs is not uniform when users choose their own PIN code, so in reality, the possibility that an attacker has to break into a random users phone in for example 5 tries is much greater than . The second most used PIN length is 6 digits, which provides much better security when using random PINs, with slightly better distribution for the most used PINs.

The mean length for a PIN is shown to be 4.82 digits long, but for PINs specifically, this value is not really representative of user choices, as there are 0 out of our 207 respondents that use a PIN code that uses a 5 digit PIN. PIN use is heavily centered around lengths of 4 or 6 digits, hence, using anything else in the input for the function will most likely yield a wrong result for most users. To get the most correct result, as well as the most optimistic results from the function, we will use 4 different levels of PIN security in the function. Those will be user-chosen 4 digit PIN, random 4 digit PIN, user-chosen 6 digit PIN and random 6 digit PIN. The entropy calculation is based on the entropy of random text, as well as the guessing entropy described in section 2. As stated in that chapter, the guessing entropy is a rough estimation, and also requires a dictionary check to eliminate for example the top 20 most used PINs, along with runs or PINs with only one number repeated.

- **User-chosen 4 digits:** 10 bits of entropy.
- **Random 4 digits:** 13.3 bits of entropy.
- **User-chosen 6 digits:** 14 bits of entropy.
- **Random 6 digits:** 19.9 bits of entropy.

### 4.4.2 Password

Both the amount of respondents who use or have previously used passwords on a phone and the length that some respondents stated they used on a mobile phone were surprising. With the small keyboard that is on mobile phones, typing long passwords could be seen as tedious, especially if

that password is complex. This reason might be why there is a relatively high amount of users not using a complex password on their phone, as it requires them to "switch" keyboards from normal letters to symbols (for iPhone users, numbers as well) while typing.

Because the distribution of lengths in passwords are not as concentrated, and both the mean and median value of password length is around 8 characters, this is the length we will use when calculating entropy for passwords. Changes to the length or adding more instances of password authentication into the ranking function can easily be done if that is necessary. As for PIN authentication,the security of both user-chosen and randomly chosen passwords will be calculated. For the entropy calculation, both passwords is assumed to be complex and ran through an extensive dictionary of bad passwords.

- **User-chosen 8 characters:** 30 bits of entropy.
- **Random 8 characters:** 52.4 bits of entropy.

### 4.4.3 Pattern

Not having any other questions about pattern authentication in our questionnaire was an oversight on our part when creating the questions. Fortunately for us, other research have been done on this authentication method, which greatly helped us. When calculating the entropy values for pattern authentication we have used the statement from [26] that a regular pattern averages a length of 5.63 dots (rounded up to 6), but only has the security equivalent to a 3 digit random PIN.

- **User-chosen 6 dots:** 9.7 bits of entropy.
- **Random 6 dots:** 14.7 bits of entropy.

## 4.5 Fingerprint recognition

The reason for why the distributions of how often users have to enter their passcode because of too many failed biometric attempts in a row is so different for iPhone and Android users might be due to algorithmic differences and a difference in where the threshold is set for Apple and Android manufacturers products affecting the FAR and FRR. The difference can also be in the sensor itself, but we deem that as unlikely as Apple offers mostly premium products, and is to our knowledge the only company that has released any documents regarding all the security of their phones (found here [36]). Despite the rather large difference in how often users have to enter their passcode after too many failed attempts between iPhones and Androids, the frequency distribution in user-friendliness scores are that different between the two operating systems.

A statistically significant Pearson correlation test showed that there were a moderate correlation between how often users have to enter their passcode after too many failed biometric attempts and the user-friendliness score. The sample sizes we have for fingerprint should also be enough for us to say with some certainty that the correlation is correct.

The security of biometric security systems is measured differently than authentication methods in the "secret" category, and this will be taken into account in the function. The measure for security we use in this thesis for biometric security systems is the FAR. Since we did not ask about how many

fingers users had enrolled in their phone, we will use the FAR of one enrolled finger. It is important to note that most likely, many users have enrolled more fingers, but we can not say so for certain, and will therefore use one finger in our calculations. If more fingers are used, the FAR of fingerprint recognition needs to be adjusted. As Apple's security document is the only document we found by a phone manufacturer on their devices fingerprint recognition security, this is the value we will be using for fingerprint recognition. Other manufacturers might have set their threshold higher or lower for their phones, thus increasing or decreasing the security of the system.

- **Fingerprint false accept rate:** 0.002%
- **Fingerprint false accept rate under targeted attack:** High

## 4.6 Facial recognition

For facial recognition, none of the users that responded said that they were locked out multiple times daily. The number of people using facial recognition is much lower than that for fingerprint recognition, and with more respondents in the facial recognition category, we might have seen someone who responded that they needed to use their passcode multiple times daily, though this is only speculation. Interestingly, the user-friendliness scores given by respondents who use an iPhone were much higher than those who use an Android phone, even though the difference in how often the users of the two different groups are locked out of their phone is not that big. When looking at the "Never" and "Rarely" cases combined for both operating systems, Android users actually had a slightly higher percentage of the answers in those categories. Regardless of this, the mean user-friendliness scores for iPhone facial recognition is a whole point higher than that of Android facial recognition. The iPhone facial recognition also has a much lower standard deviation at 0.855 compared to 1.32 for Android, and is statistically significant on the 5% level. This large difference in user-friendliness might be due to the neural networks iPhones use to learn how the users face looks under different circumstances like with/without glasses, hats, scarfs, beard etc which can give an added feeling of user-friendliness.

The correlation test showed a much stronger relationship between how often a passcode is needed and the user-friendliness scores for Android phones than for Apple. The correlation test for Android phones were significant at the 1% level, while the correlation on iPhones were significant at the 5% level.

Like with fingerprint, Apple's security document is the only document we found by a phone manufacturer on their devices facial recognition security, so this is the value we will be using for 3D facial recognition. Other manufacturers might have set their threshold higher or lower for their phones, thus increasing or decreasing the security of the system. For 2D facial recognition, no such document was found, but with how easy successfully attacking these system is shown to be, there is no real security in using such an authentication system. The only time such a system would be of any use is in scenarios where the attacker has no way of finding out who the owner of the phone is, for example when finding a random phone on the street with no way of finding the identity of the phone's owner. Of course, some phones using 2D facial recognition might be more secure than for

example the Galaxy S10, but as we have not tested any more phones using 2D facial recognition, we do not feel comfortable suggesting the use of such a system knowing how easy and effortless breaking into the Galaxy S10 is.

- **2D facial recognition:** Should not be used in any secure setting.
- **3D false accept rate:** 0.0001%
- **3D false accept rate under targeted attack:** High

## 4.7 Iris recognition

The 9 answers we got from the questionnaire on iris recognition is too low of an amount to really draw anything conclusive from the answers. As stated earlier, we will nonetheless use the user-friendliness result, as it at least gives us a pointer to how user-friendly a mobile phone iris recognition security system is. With the user-friendliness score being calculated to be 3.22, it is placed last on the list of our suggested authentication methods. The score is as far from the next authentication method on the list (passwords) as that authentication method is to the most user-friendly authentication method (facial recognition).

As we have not found any specific numbers for the false accept rate for a phone that is capable of iris recognition, how we set the security will be an estimation based on other information on iris recognition. Samsung claims that the iris recognition on their phones (for the phones that offer it) is the most secure way of authenticating yourself. As we have assumed the fingerprint FAR to be 0.002%, the iris recognition will then be set to have a smaller FAR. The scanner in their phones uses near infrared light (NIR) to better enhance the pattern in the iris, as other good sensors do. Mark Clifton, CEO of Princeton Identity, the company that Samsung partnered with to implement their iris scanners said to Business Insider that the Galaxy S8 registers up to 200 identifying features per iris [70]. Because of the alleged high security and scanner using best practice methods to acquire the iris image, we decided that we will set the FAR for iris recognition to the same as 3D facial recognition. It is important to note that this number will most likely not represent the actual FAR of the iris recognition system.

- **Iris false accept rate:** 0.0001%
- **Iris false accept rate under targeted attack:** High

## 4.8 Multifactor authentication and use of varying steps of security

When users opt for biometric authentication on their phones, they also need to create a PIN, password or pattern in case the biometric authentication method fails to authenticate the user too many times in a row. Because of this, no matter how secure a biometric authentication method is, if the PIN code on the phone is set to be "1234", the actual security of the phone is near non-existent, as the weakest link in the authentication process will still be the PIN for this case. The PIN/password/pattern is also the only way to change or turn off authentication and the lock screen and other important aspects of the security of the phone itself, so these authentication methods should be made at least secure enough to withstand guessing and dictionary attacks. If the passcode is

made to be strong enough, the biometric method of choice can then be used on a day-to-day basis as they are shown to mostly be more user-friendly than passcodes, while the strong passcode is used when biometric authentication fails.

Since the passcode authentication always needs to exist on phones, it might also be easier for users to accept using the biometric authentication method and their passcode in cases where two-factor authentication is needed, as they have to remember the passcode anyway, and it doesn't add anything more to remember for users. It is also possible to use varying steps of security within for example an app. An example of this can be a banking app, where the user is allowed to log in with a PIN to see their balance and transfer funds across their own accounts, but the user wants to make a payment to another person, the user will need to authenticate themselves further by using specifying their national identity number, a one-time code sent to either a dedicated token or the phone and a a longer password.

## 4.9 Limitations

There are some limitations to using an online survey to collect answers from users. One such limitation is that we cannot verify the honesty and accuracy of the respondents answers. Because we are asking for information about peoples way of authenticating themselves to their phones, we can't ask for their names and such for ethical reasons. Even though we would never use any such identifying information for personal gain by for example breaking into a respondents phone with the information the respondent gave up in the survey, if that information got out of our hand it could pose a security threat to those who responded. We chose to use a questionnaire to avoid the manual work and to keep the people who participated anonymous.

Another limitation is that user-friendliness is inherently subjective. How the respondents think of user-friendliness and the reasoning behind their given user-friendliness score can not be verified. Misreading the text can also lead to wrong answers being entered with no way for us to verify if it is what the subject really feels or not. An example of this from our dataset is when a respondent gave authenticating with password a user-friendliness score of 6, while giving fingerprint a score of 3. This gets even more peculiar when the password is stated to be 9 characters long and complex. This could be due to the fact that the respondent might have problems with getting access using the fingerprint. If that happens a lot then it is annoying for the user and the method will be considered not user-friendly. If the password is relatively simple to enter and the person is used to entering it correctly, then this might be seen as user friendly.

A limitation to the performance of our function as that because we haven't been able to verify the accuracy and thresholds for many of the biometric authentication methods, the ranked outcome can be wrong. Because some authentication methods are still not that popular in use, their user-friendliness scores can also be easily skewed by just a few users.

# 5 Conclusion and future work

## 5.1 Conclusion

In this thesis we have created a function that ranks authentication methods based on their security and user-friendliness. The security measurements were collected from literature, while a questionnaire was issued to collect answers on user-friendliness. The attack we performed against the Galaxy S7 Edge's fingerprint recognition were mostly unsuccessful, while the attack against the Galaxy S10's facial recognition were mostly successful.

**Main research question - The best authentication method on mobile devices**
The result of running the implementation of the function show that for a mobile security system, facial recognition is the best authentication method when considering both security and user-friendliness.

**Secondary research question 1 - Measuring security**
To measure security we opted for using entropy for password systems and the false accept rate for biometric systems. When creating the ranked list, they were converted to the same 1-6 scale as the user-friendliness was measured in so it would be possible to calculate the total score for both types of authentication methods.

**Secondary research question 2 - Measuring user-friendliness**
Measuring user-friendliness was The measure of user-friendliness could have benefited from being more thoroughly research by us asking more questions in the questionnaire. What we did get, is a crude number measurement of how users feel about different authentication methods. We also saw that the user-friendliness score do not differ much for most of the authentication methods. The most user-friendly method was shown to be facial recognition.

## 5.2 Future work

As some of the authentication methods we suggested in our questionnaire got very few answers, it would be beneficial to at least have these methods get more answers so that the mean user-friendliness result gets more representative. The questionnaire could also be altered to have more questions that could be interesting, for example how secure users think different authentication methods are and questions about the ease of use, intuitiveness, speed and reliability to get a better understanding of the reasoning behind the user-friendliness score given.

The closer the security numbers used when creating the ranked list are to reality, the more nuanced and accurate the results will be as well. Because most phone manufacturers are not open about the performance of their biometric authentication methods, a lot of assumptions had to be made. Lowering the amount of assumption will raise the credibility level of the outcome.

# Bibliography


[1] Three finger prints isolated on white. accessed: 30.05.2019. URL: https://stock.adobe.com/no/images/three-finger-prints-isolated-on-white/4262071?asset_id=4262071.

[2] arch. accessed: 30.05.2019. URL: https://stock.adobe.com/no/images/arch/315272?asset_id=315272.

[3] fingerprint vector illustration.fingerprint scan. accessed: 30.05.2019. URL: https://stock.adobe.com/no/images/fingerprint-vector-illustration-fingerprint-scan/192705433?asset_id=192705433.

[4] Impronta digitale. accessed: 30.05.2019. URL: https://stock.adobe.com/no/images/impronta-digitale/154988873?asset_id=154988873.

[5] Müller, R. 2001. In *Fingerprint Verification with Microprocessor Security Tokens*, 27–41.

[6] Berry, N. Pin analysis. accessed: 23.05.2019. URL: http://www.datagenetics.com/blog/september32012/index.html.

[7] William E. Burr, Donna F. Dodson, E. M. N. R. A. P. W. T. P. 2013. Electronic authentication guideline.

[8] Løge, M. D. Tell me who you are and i will tell you your unlock pattern. Master's thesis, NTNU, 2015.

[9] Jain, A. K., Ross, A., & Prabhakar, S. 2004. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1), 4–20.

[10] Patel, K., Han, H., & Jain, A. K. 2016. Secure face unlock: Spoof detection on smartphones. *IEEE transactions on information forensics and security*, 11(10), 2268–2283.

[11] Helkala, K. & Snekkenes, E. 2009. Formalizing the ranking of authentication products. *Information Management & Computer Security*, 17(1), 30–43.

[12] Malone, D. & Maher, K. 2012. Investigating the distribution of password choices. In *Proceedings of the 21st international conference on World Wide Web*, 301–310. ACM.

[13] Mazurek, M. L., Komanduri, S., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., Kelley, P. G., Shay, R., & Ur, B. 2013. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 173–186. ACM.

[14] Bonneau, J. 2012. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Security and Privacy (SP), 2012 IEEE Symposium on*, 538–552. IEEE.

[15] Bonneau, J. & Shutova, E. 2012. Linguistic properties of multi-word passphrases. In *International Conference on Financial Cryptography and Data Security*, 1–12. Springer.

[16] Vance, A. 2010. If your password is 123456, just make it hackme. *The New York Times*, 20, A1–A1.

[17] Ur, B., Komanduri, S., Shay, R., Matsumoto, S., Bauer, L., Christin, N., Cranor, L. F., Kelley, P. G., Mazurek, M. L., & Vidas, T. 2013. Poster: The art of password creation. In *Proc. IEEE Symposium on Security and Privacy*.

[18] Ur, B., Noma, F., Bees, J., Segreti, S. M., Shay, R., Bauer, L., Christin, N., & Cranor, L. F. 2015. I added '!'at the end to make it secure": Observing password creation in the lab. In *Proc. SOUPS*.

[19] Imperva, A. Consumer password worst practices. Technical report, Technical report, Imperva ADC, 2010. URL: https://www.imperva.com/docs/gated/WP_Consumer_Password_Worst_Practices.pdf.

[20] Veras, R., Thorpe, J., & Collins, C. 2012. Visualizing semantics in passwords: The role of dates. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, 88–95. ACM.

[21] Veras, R., Collins, C., & Thorpe, J. 2014. On semantic patterns of passwords and their security impact. In *NDSS*.

[22] Florencio, D. & Herley, C. 2007. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, 657–666. ACM.

[23] Grassi, P., Fenton, J., Newton, E., Perlner, R., Regenscheid, A., Burr, W., Richer, J., Lefkovitz, N., Danker, J., Choong, Y.-Y., et al. 2017. Nist special publication 800-63b: Digital identity guidelines.

[24] Chell, D. Apple ios hardware assisted screenlock bruteforce. accessed: 07.05.2019. URL: http://blog.mdsec.co.uk/2015/03/bruteforcing-ios-screenlock.html.

[25] De Angeli, A., Coventry, L., Johnson, G., & Renaud, K. 2005. Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems. *International journal of human-computer studies*, 63(1-2), 128–152.

[26] Uellenbeck, S., Dürmuth, M., Wolf, C., & Holz, T. 2013. Quantifying the security of graphical passwords: the case of android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 161–172. ACM.

[27] Berry, J. & Stoney, D. A. 2001. The history and development of fingerprinting. *Advances in fingerprint Technology*, 2, 13–52.

[28] Cole, S. A. 2004. History of fingerprint pattern recognition. In *Automatic Fingerprint Recognition Systems*, 1–25. Springer.

[29] Ratha, N. K., Connell, J. H., & Bolle, R. M. 2001. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3), 614–634.

[30] Roy, A., Memon, N., & Ross, A. Sept 2017. Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems. *IEEE Transactions on Information Forensics and Security*, 12(9), 2013–2025. doi:10.1109/TIFS.2017.2691658.

[31] darkshark. I attempted to fool the new samsung galaxy s10's ultrasonic fingerprint scanner by using 3d printing. i succeeded. accessed: 30.04.2019. URL: https://imgur.com/gallery/8aGqsSu#WlRksn6.

[32] Mayhew, S. German researcher reverse-engineers a fingerprint using photos. accessed: 22.05.2019. URL: https://www.biometricupdate.com/201412/german-researcher-reverse-engineers-a-fingerprint-using-photos.

[33] Hern, A. Hacker fakes german minister's fingerprints using photos of her hands. accessed: 22.05.2019. URL: https://www.theguardian.com/technology/2014/dec/30/hacker-fakes-german-ministers-fingerprints-using-photos-of-her-hands.

[34] starbug: Ich sehe, also bin ich ... du (english translation). accessed: 22.05.2019. URL: https://www.youtube.com/watch?v=VVxL9ymiyAU&feature=youtu.be.

[35] Jain, R. & Kant, C. 09 2015. Attacks on biometric systems: An overview. *International Journal of Advances in Scientific Research*, 1, 283. doi:10.7439/ijasr.v1i7.1975.

[36] Inc.", A. May 2019. ios security ios 12.3.

[37] Jain, A. K. & Li, S. Z. 2011. *Handbook of face recognition*. Springer.

[38] Therapy", U. Samsung galaxy s10 unlock hack (warning). accessed: 20.05.2019. URL: https://www.youtube.com/watch?v=BGgQ9woZQOg&t=205s.

[39] dot Com", J. C. How to crack samsung galaxy s10 facial recognition. accessed: 20.05.2019. URL: https://www.youtube.com/watch?v=1mdnR6OsUjM.

[40] Ulanoff, L. I spoofed the samsung galaxy s10+ facial recognition with a photo. accessed: 20.05.2019. URL: https://www.youtube.com/watch?v=Ta8sHX-wLTk.

[41] bkav.com. november 2017. Bkav's new mask beats face id in "twin way": Severity level raised, do not use face id in business transactions. Accssed: 10.11.2018. URL: http://www.bkav.com/d/top-news/-/view_content/content/103968/bkav%92s-new-mask-beats-face-id-in-twin-way-severity-level-raised-do-not-use-face-id-in-business-t

[42] Xu, Y., Price, T., Frahm, J.-M., & Monrose, F. 2016. Virtual u: Defeating face liveness detection by building virtual models from your public photos. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 497–512.

[43] Pan, G., Sun, L., Wu, Z., & Lao, S. 2007. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *2007 IEEE 11th International Conference on Computer Vision*, 1–8. IEEE.

[44] Chingovska, I., Anjos, A., & Marcel, S. 2012. On the effectiveness of local binary patterns in face anti-spoofing. In *2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*, 1–7. IEEE.

[45] Raja, K. B., Raghavendra, R., & Busch, C. 2015. Video presentation attack detection in visible spectrum iris recognition using magnified phase information. *IEEE Transactions on Information Forensics and Security*, 10(10), 2048–2056.

[46] Tirunagari, S., Poh, N., Windridge, D., Iorliam, A., Suki, N., & Ho, A. T. 2015. Detection of face spoofing using visual dynamics. *IEEE transactions on information forensics and security*, 10(4), 762–777.

[47] de Freitas Pereira, T., Anjos, A., De Martino, J. M., & Marcel, S. 2013. Can face anti-spoofing countermeasures work in a real world scenario? In *2013 international conference on biometrics (ICB)*, 1–8. IEEE.

[48] Menotti, D., Chiachia, G., Pinto, A., Schwartz, W. R., Pedrini, H., Falcao, A. X., & Rocha, A. 2015. Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Transactions on Information Forensics and Security*, 10(4), 864–879.

[49] Boulkenafet, Z., Komulainen, J., & Hadid, A. 2015. Face anti-spoofing based on color texture analysis. In *2015 IEEE international conference on image processing (ICIP)*, 2636–2640. IEEE.

[50] Arashloo, S. R., Kittler, J., & Christmas, W. 2015. Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features. *IEEE Transactions on Information Forensics and Security*, 10(11), 2396–2407.

[51] Kim, W., Suh, S., & Han, J.-J. 2015. Face liveness detection from a single image via diffusion speed model. *IEEE transactions on Image processing*, 24(8), 2456–2465.

[52] Wen, D., Han, H., & Jain, A. K. 2015. Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security*, 10(4), 746–761.

[53] Galbally, J., Marcel, S., & Fierrez, J. 2013. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE transactions on image processing*, 23(2), 710–724.

[54] Pinto, A., Pedrini, H., Schwartz, W. R., & Rocha, A. 2015. Face spoofing detection through visual codebooks of spectral temporal cubes. *IEEE Transactions on Image Processing*, 24(12), 4726–4740.

[55] Zhang, Z., Yi, D., Lei, Z., Li, S. Z., et al. 2011. Face liveness detection by learning multispectral reflectance distributions. In *FG*, 436–441.

[56] Samsung.com. Use facial recognition security on your phone. accessed: 23.05.2019. URL: https://www.samsung.com/us/support/answer/ANS00062630/.

[57] Kücken, M. & Newell, A. C. 2005. Fingerprint formation. *Journal of theoretical biology*, 235(1), 71–83.

[58] Daugman, J. 2003. The importance of being random: statistical principles of iris recognition. *Pattern recognition*, 36(2), 279–291.

[59] Samsung.com. Set up a screen lock on your phone. accessed: 25.05.2019. URL: https://www.samsung.com/us/support/answer/ANS00078955/.

[60] Farooqui, A. Samsung said to ditch the galaxy s10 iris scanner. accessed: 25.05.2019. URL: https://www.sammobile.com/2018/11/02/samsung-ditch-galaxy-s10-iris-scanner/.

[61] Bowyer, K. W. & Burge, M. J. 2016. *Handbook of iris recognition*. Springer.

[62] Steve_At_Hypothermia. Hacking the samsung galaxy s8 irisscanner. accessed: 25.05.2019. URL: https://www.youtube.com/watch?time_continue=72&v=H9pfc_4cNBc.

[63] Trenner, L. 1987. How to win friends and influence people: definitions of user-friendliness in interactive computer systems. *Journal of information science*, 13(2), 99–107.

[64] O'Gorman, L. 2003. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12), 2021–2040.

[65] Ross, A., Nandakumar, K., & Jain, A. K. 2008. Introduction to multibiometrics. In *Handbook of biometrics*, 271–292. Springer.

[66] Shams. List of all eye scanner (iris, retina recognition) smartphones. accessed: 03.05.2019. URL: https://webcusp.com/list-of-all-eye-scanner-iris-retina-recognition-smartphones/.

[67] Adler, A., Youmaran, R., & Loyka, S. 2006. Towards a measure of biometric information. In *2006 Canadian Conference on Electrical and Computer Engineering*, 210–213. IEEE.

[68] Takahashi, K. & Murakami, T. 2018. A generalization of the theory of biometric system entropy. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1–6. IEEE.

[69] Bera, A. How to fix galaxy note 8 iris scanner issues. accessed: 29.05.2019. URL: https://www.technobezz.com/fix-galaxy-note-8-iris-scanner-issues/.

[70] Villas-Boas, A. The company behind the galaxy s8's iris recognition says it's superior to the fbi's fingerprint tech. accessed: 29.05.2019. URL: https://www.businessinsider.com/samsung-galaxy-s8-iris-scanner-fbi-fingerprint-tech-princeton-identity-2017-4?r=US&IR=T.

# A Questionnaire

1. What is your gender?
   - Male
   - Female
2. How old are you?
   - 15-24
   - 25-34
   - 35-44
   - 45-54
   - 55-64
   - 65+
3. What kind of phone do you have?
   - iPhone
   - Android
   - Other
4. Do you use a lockscreen on your phone?
   - Yes
   - No
5. Why are you not using a lockscreen?
   - Open answer
6. What are you using now or have used before? Multiple choices is possible
   - PIN
   - Password
   - Pattern
   - Fingerprint recognition
   - Facial recognition
   - Iris recognition
   - Other
7. How many digits are in your PIN-code?
   - 4
   - 5

- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- Don't know

8. A complex password is a password that contains at least one lower case letter, one upper case letter, a digit and a special character. Are you using a complex password?

   - Yes
   - No

9. How many characters is in your password?

   - 4
   - 5
   - 6
   - 7
   - 8
   - 9
   - 10
   - 11
   - 12
   - 13
   - 14
   - 15
   - 16+

10. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of a PIN-code is?

    - 1
    - 2
    - 3
    - 4
    - 5
    - 6

11. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of a password is?
    - 1
    - 2
    - 3
    - 4
    - 5
    - 6
12. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of a pattern is?
    - 1
    - 2
    - 3
    - 4
    - 5
    - 6
13. How often do you encounter being rejected by your phone so many times that fingerprint gets deactivated and you have to use your PIN to log in?
    - Never
    - Rarely
    - Monthly
    - Weekly
    - Daily
    - Multiple times daily
14. How often do you encounter being rejected by your phone so many times that facial recognition gets deactivated and you have to use your PIN to log in?
    - Never
    - Rarely
    - Monthly
    - Weekly
    - Daily
    - Multiple times daily
15. How often do you encounter being rejected by your phone so many times that iris recognition gets deactivated and you have to use your PIN to log in?
    - Never
    - Rarely

- Monthly
- Weekly
- Daily
- Multiple times daily

16. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of "other" is?

    - 1
    - 2
    - 3
    - 4
    - 5
    - 6

17. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of fingerprint recognition is?

    - 1
    - 2
    - 3
    - 4
    - 5
    - 6

18. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of facial recognition is?

    - 1
    - 2
    - 3
    - 4
    - 5
    - 6

19. On a scale from 1-6, where 1 is lowest and 6 is highest, how user-friendly do you think the use of iris recognition is?

    - 1
    - 2
    - 3
    - 4
    - 5
    - 6

# B CSV file

method,type,security,friendliness,note
PIN user-chosen 4 digits,secret,10,4.33,"Intensive dictionary check is needed, without it, security will be lower"
PIN random 4 digits,secret,13.3,4.33,none
PIN user-chosen 6 digits,secret,14,4.33,"Intensive dictionary check is needed, without it, security will be lower"
PIN random 6 digits,secret,19.9,4.33,none
Password user-chosen 8 char,secret,30,3.99,"Intensive dictionary check is needed, without it, security will be lower"
Password random 8 char,secret,52.4,3.99,none
Pattern user-chosen 6 dots,secret,9.7,4.45,none
Pattern random 6 dots,secret,14.7,4.45,none
Fingerprint,biometric,0.002,4.66,High FAR when under targeted attack by capable and motivated attacker
2D face,biometric,0.015,4.2,"Very high FAR even from low effort attacks, and should not be used in any kind of secure setting"
3D face,biometric,0.0001,5.21,High FAR when under targeted attack by capable and motivated attacker
Iris,biometric,0.0001,3.22,High FAR when under targeted attack by capable and motivated attacker

NTNU
Norwegian University of
Science and Technology

buypass