One-Time Password Security

IN2120 Information Security Home Exam

Ahlam Aatif & Zixuan Liu

Candidate numbers respectively: 15287 & 15215

Department of Informatics

UNIVERSITY OF OSLO

04.11.2018

Abstract

One-time password (OTP) is a modern industry standard for authentication systems requiring enhanced security. It is often used in combination with two-factor authentication, in sectors such as online banking, eHealth, national defense and law enforcement. OTP-based systems are, nonetheless, vulnerable to security breaches. In the past decade, multiple banks, websites, mobile applications and even authentication security companies have been victims of security attacks. This paper analyzes the mechanisms behind OTPs, identifies the most common attacks towards OTP-based systems, conducts a short experiment on popular mobile applications, and suggests risk mitigation strategies for each identified risk.

Keywords: One-Time Passwords, OTP Vulnerabilities, OTP Security

Table of Contents

1. Introduction 5
1.1 Motivation and presentation of problem 5
1.2 Terminology 6
1.3 Structure 6
1.4 References 6
1.5 Limitations 6

2. Background 7
2.1 Introduction to One-Time Passwords 7
2.2 Uses of One-Time Passwords 7
2.3 One-Time Password Generation 8
2.3.1 Asynchronous one-time passwords 8
2.3.1.1 Code books 8
2.3.1.2 Challenge-response System 9
2.3.2 Synchronous One-Time Passwords 9
2.3.2.1 Counter-based 9
2.3.2.2 Time-based 10
2.4 Delivering One-Time Passwords 11
2.4.1 Hard Copy 11
2.4.2 SMS 11
2.4.3 Email and Mobile Applications 11
2.4.4 Proprietary Token Devices 11

3. OTP Security Vulnerabilities 12
3.1 Physical Loss & Theft 12
3.2 Social Engineering 13
3.2.1 Mass frauds 13
3.2.2 Targeted Fraud 14
3.3 Trial-and-error Password Cracking 14
3.3.1 Brute-force Attack 15
3.3.2 Dictionary Attack 16
3.3.3 Rainbow Table attack 16
3.3.4 A short experiment: Overview and methodologies 16
3.3.5 Presentation of data 17
3.3.6 Discussion of results 17
3.4 Man-in-the-Middle (MITM) Attacks 18
3.5 Trojan Horse Attack 18

4. Suggestions to security measures 19
4.1 Protection Against Loss & Theft 19

4.2 Protection Against Phishing 20
4.3 Protection Against Trial-and-error Password Cracking 21
4.4 Protection Against Man-in-the-Middle attack 22
4.5 Protection Against Trojan Horse Attacks 23

5. Conclusion 24

6. Bibliography 25

6.1 Companies mentioned in this paper 29

1. Introduction

1.1 Motivation and presentation of problem

On March 17, 2011, RSA Security publicly announced through an open letter that "an extremely sophisticated cyber attack" was being mounted against the company (Art Coviello, u.y). According to the letter, sensitive information related to the company's SecurID two-factor authentication products were extracted from RSA's systems, thus weakening the protection offered by these products. As a result, the security breach costed RSA's parent company, EMC, an estimated total of $66.3 million. Additionally, there were numerous attacks on RSA's customers, the most notable being Lockheed Martin - one of the largest security, technology and aerospace corporations in the world (Eric Chabrow, 2011).

The RSA SecurID authentication system is a well known example for time-based one-time passwords, also known as a **synchronous** one-time password (Harris & Maymí, 2016. p. 756). There are many other examples of one-time password devices that fall into this category, some examples being Yubico's YubiKey and the Norwegian authentication system BankID. In Norway, BankID is classified as the most secure level of authentication by the Norwegian Framework for Authentication and Non-Repudiation, is used by approximately four million Norwegians, for everything from completing bank transactions to accessing official documents (Regjeringen, 2008).

As a result of the rising security threats, an increasing amount of common web- and mobile applications are also changing from static to dynamic passwords.

The motivation behind our topic choice is therefore quite simple. One-time password, especially when combined with two-factor authentication, is assigned a high level of trust and adopted by users and organizations worldwide. However, they have obvious vulnerabilities as illustrated by the case of RSA security breach. This combination makes OTP security an extremely interesting topic for in-depth explorations. In this paper, we will therefore examine the following research question:

What are some critical security threats for authentication systems using
One-Time Passwords (OTPs), and how can one protect against these threats?

1.2 Terminology

One-time password is here defined as a password that expires after a single authentication. The term "one-time password" is also known as *dynamic password*, and often interchanged with the abbreviation "OTP" in this paper. It is important to distinguish OTP from *one-time* *pads*, which is an encryption technique also mentioned in this paper. Secondly, the paper mentions security token devices used to deliver the one-time passwords. *Security token device* is here defined as “a device that generates the one-time password for the user to submit” (Harris & Maymí, 2016. p.754). Finally, one should note the difference between the words “authentication” and “authorization”. *Authorization* is when the function specifies the access privileges an user should receive, basing on the role of the user. This term is often mistaken as access control (Audun Jøsang, 2018). Authentication, on the other hand, is defined by the IETF as “the process of verifying a claim that a system entity or system resource has a certain attribute value” (IETF, 2007. p.26).

1.3 Structure

The paper is divided into three sections. The first section provides a general presentation of OTPs, a comparison of OTPs with traditional static passwords, as well as various generation and delivery methods for OTPs. The second part explores significant security threats to OTP-based authentication systems, sorted in ascending order by their relative technical complexity. There will be real-life examples alongside each threat scenario to illustrate the risk of the threat, as well as a short experiment on popular mobile applications. Finally, the last chapter suggests risk mitigation strategies to each identified threat.

1.4 References

All definitions in this paper comes from RFC4949 - *Internet Security Glossary, Version 2* published by the Internet Engineering Task Force (IETF), an international organization developing open internet standards. As OTP security covers a broad spectrum of topics in information security, a wide range of scientific papers have been referenced. Many of the measures against security threats of OTPs are based on requirements and recommendations from the ISO27001:2013, ISO27002:2013 and ISO29115:2012, as they contain an international standard for information security.

1.5 Limitations

The scope of this home exam leads to some unavoidable limitations. Firstly, one-time password security is a broad topic covering almost all spectrums of information security, and we were unable to explain every concept in detail. The list of threats for OTPs is non-exhaustive, and the time and length constrictions of this project limit the number of threats and solutions we were able to examine and experiment. We have therefore chosen to prioritize the most critical security threats and protection methods against these threats, and specifically focus on brute-force attacks in our short experiment.

Secondly, legal and ethical considerations limit the extent to which hacking methods, as well as our self-written codes, could be tested on the applications and websites. We have therefore mentioned real-life applications with potential security flaws, without actually testing and breaking the system, and rather presented proof-of-concepts.

Lastly, due to the broad scope, lack of a specific product and limited resources, we have decided to refrain from threat modeling. Instead, we have chosen to explore the most common security threats for OTP-based systems.

2. Background

2.1 Introduction to One-Time Passwords

Most authentication systems can be classified into three categories, commonly known as the three factors of authentication - something the user knows, something the user has and something the user is or does. Research has shown that for positive authentication, elements from at least two of the factors should be verified, giving rise to the term two-factor authentication (Federal Financial Institutions Examination Council, u.y). A traditional, static username and password combination falls into “something the user knows”, whereas an OTP commonly categorizes as “something the user has”. One-time password is therefore often used in two-factor authentication systems, where the user must be verified with both a static password and an OTP.

A one-time password is meant for single-use only. Once the authentication has taken place, the password expires and is rendered useless to the attacker. As a result, OTPs are far more resistant towards replay attacks than traditional static passwords, where the attacker can capture the password, and resubmit it with the intent of accessing unauthorized information (Harris & Maymí, 2016. p.413).

2.2 Uses of One-Time Passwords

Nowadays, OTPs with two-factor authentication are commonly used in settings requiring enhanced security. Some notable examples are the majority of online banking industries, online health industries, as well as systems connected to national defense and law enforcement (Alcodes, 2016).

Additionally, many digital ecosystem giants, such as Google, Apple and Microsoft, offer an OTP-based two-factor authentication system for the users. The authentication system can verify the user both during a normal login session, but also upon retrieving forgotten passwords, after failed login attempts, during attempts of login through new devices and locations, and when logging in following a period of inactivity.

OTP can also be used as a standalone single-factor authentication tool. This has been adopted by popular mobile applications, examples being the dating app Tinder, the instant messaging app WhatsApp, the Chinese WeChat, the 7-Eleven App and the Espresso House App.

In Norway, The Norwegian Data Protection Authority (DPA) requires a “strong authentication system” for all users with access to information systems containing sensitive personal information. This is exemplified by the use of a two-factor authentication system including an OTP and a static username-password combination (Datatilsynet, u.y). Additionally, both the Norwegian Framework for Authentication and Non-Repudiation and ISO29115:2012 - the international standard for Entity Authentication Framework, specifies four levels of authentication assurance, in which the two highest levels require multi-factor verification (Regjeringen, 2008). OTP-based two-factor authentication systems satisfy the highest level of of authentication assurance.

2.3 One-Time Password Generation

One-time passwords can in general be divided into two categories, basing on their generation method - an asynchronous one-time password, or a time-dependent, *synchronous* one-time password (Harris & Maymí, 2016, p.754). Synchronous OTPs are in general more secure than asynchronous OTPs, as they are additionally protected by their time- and counter-dependent property, and is most widely in use today. A common ground for any OTP generation algorithm is their *pseudo-randomness*. This prevents the attacker from predicting future passwords.

2.3.1 Asynchronous one-time passwords

2.3.1.1 Code books

One way to generate one-time passwords is by generating an entire code book based on a shared secret. This is often generated using a type of one-time pad (UNIX and Internet Security, u.y). A one-time pad is an encryption algorithm, in which the key is a random sequence of symbols, and each symbol is only used once for encryption. A copy of the key is used similarly for decryption (IETF, 2007, p.205).

When a user wishes to log into a system, they must either look up the next password in the code book or generate the next password in a virtual codebook. This method was popular among many European online banks in the 2000’s, an example being Nordea.

2.3.1.2 Challenge-response System

Another method to generating asynchronous one-time passwords is by presenting the user with a challenge when attempting to log in. This challenge is a random value, also called a nonce, which the user must enter into the token device. The token device then returns a value, which serves as an OTP. The user must then enter the OTP to the authentication server, along with a username, as a response to the challenge (Harris & Maymí, 2016. p.756). Successful authentication requires the user's response to match the response of the host server.

2.3.2 Synchronous One-Time Passwords

Synchronous OTP generation is based on the synchronization of an authentication server and a token device, using time or a counter as the central piece in the authentication process (Harris & Mamí, 2002. p.755).

2.3.2.1 Counter-based

A counter-based OTP generation method is also known as an event-based method. In a counter-based OTP generation method, the user must initiate the creation of the OTP. This prompts both the token and the authentication server to update an internal counter in synchronization. The counter functions as an authentication value, and this value, in addition to a shared secret key, are used to generate an OTP through an algorithm (Ibid).

A common algorithm for counter-based OTP generation is called the HMAC (Hashed Message Authentication Code) based One-Time Password, abbreviated as *HOTP*. This algorithm was published by the IETF in the RFA226 documentation, and is divided into several steps (IETF, 2005):

1. Generate a hashed message authentication code *HMAC* from the secret key *K* and counter *C*, with hash function *H*

$$HMAC = HMAC_H(K, C)$$

2. This function returns a 160-bit code, which must be *truncated* for the user

$$HOTP = truncate(HMAC) = truncate(HMAC_H(K, C))$$

3. Finally, the truncated HOTP must be shortened to a desired number of digits *d*, using the *modulo* operator:

$$HOTP = HOTP \bmod 10^d$$

   If the original HOTP is 12345678 and the desired number of digits is 4, the new HOTP will be 12345678 mod 10^4 = 5678.

4. This gives an overall algorithm:

$$HOTP = truncate(HMAC_H(K, C)) \bmod 10^d$$

A major challenge in counter-based OTP generation is the risk for delays in counter updates. This is solved by using a time-based OTP generation method.

2.3.2.2 Time-based

In a time-based OTP authentication system, both the host server and the token device must contain an internal clock. The time held by the clock now functions as the counter from the previous sub-chapter. For a successful authentication, the clock must show the same time as the clock in the authentication server, although most systems tolerate minor deviations (Audun Jøsang, 2018). Both parties will then generate the same OTP, using the time value, user ID and a secret key. The process can be seen in Figure 1.

The algorithm used to generate the OTP is called the *Time-based One-Time Password Algorithm (TOTP)*, published in the RFC6238 document by IETF (IETF, 2011, Section 4). TOTP is an extension of the HOTP algorithm covered in the previous chapter (section 2.3.2.1), the only notable difference being time step *T* replacing the counter *C*. To avoid time-zone errors, the algorithm uses the *Unix timestamp*, which is independent of time-zones.

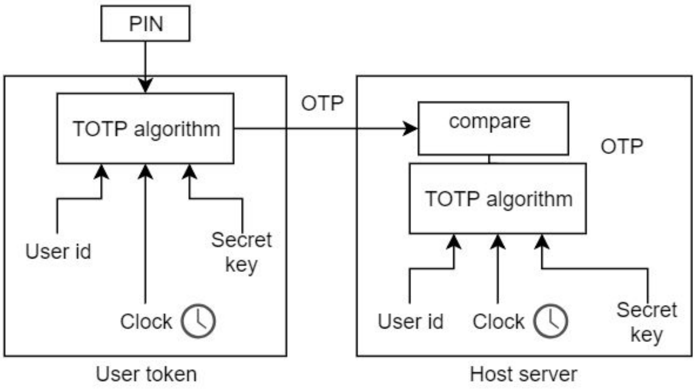


Figure 1 - Time-based OTP Token device with optional input PIN

Since TOTP is based on the Unix timestamp, the variable *T* will change for each second, meaning a new OTP is generated for each passing second. This can create severe synchronization problems, and to solve this problem, the algorithm counts *time steps* instead of the actual time. A time step size of 30 seconds means the system allows a 30-second deviation between the user token and the host server, before the OTP expires. A larger time step size allows for greater usability, but exposes a larger window to attack (IETF, 2011. Section 5.2).

2.4 Delivering One-Time Passwords

One-time passwords can be delivered to the user using different methods, through different mediums. The most common OTP delivery methods will be explored in this chapter.

2.4.1 Hard Copy

In many countries, banks generate a numeric list of OTPs that are printed on a sheet of physical paper, generated using the code book method (section 2.3.1.1). Other banks use a scratchcard to reveal an OTP. This OTP delivery method is called distribution via hard copies. MinID, an electronic ID in Norway, uses both SMS and hard copies as delivery method. If the user logs in for the very first time, they can make use of hard copies. After their first login session, they are able to activate OTP delivery through the phone. Hard copies can either be delivered to the user's registered post address, or delivered in hand to the user, depending on the security level of the authentication system.

2.4.2 SMS

One of the most common OTP delivery methods is through Short Message Service (SMS), commonly called text messaging. Text messaging is an omnipresent communication channel, as most users carry a cell phone which supports SMS at all times. This method does not require internet access, rather it only employs GSM (Global System for Mobile Communications) to send the OTP. As a result, text messages have the potential to reach a wide range of users. Most two-factor authentication systems involving OTP uses SMS as a delivery method, and many mobile applications use SMS as a standalone single-factor authentication system.

2.4.3 Email and Mobile Applications

Occasionally, OTPs can be distributed to the users via email or in a mobile application. The user can either receive an email with an OTP, or generate an OTP by initiating the authentication on a mobile application. An example of such an application is Google's Google Authenticator, which employs the time-based OTP generation method as explained in section 2.3.2.2 (Google Play, 2017).

This can be less secure than the SMS method, as it is generally recommended that two-factor authentication systems employ two different channels of communication in the authentication process. In the case of SMS OTP delivery, the primary channel is the internet, whereas the the secondary channel is the mobile network.

2.4.4 Proprietary Token Devices

A proprietary token device for OTPs is usually a handheld device that has a LCD display, and possibly a keypad (Harris & Maymí, 2016. p.754). The user can use this device to initiate the authentication process by pressing a button, and to view the generated OTP. Some tokens require the user to enter a correct username and password combination in order to access the OTP, thus creating a two-factor authentication system.

Most devices do not require any physical or logical connection to the client computer. This type of device is called *disconnected tokens*. Other devices may require physical connection to the computer, for example through a USB port, and these devices are called *connected tokens*. Finally, some devices do not require physical connection, but must be logically connected. They are called *contactless tokens* (Siemens, 2007). Examples of proprietary token devices are the Norwegian BankID and RSA's SecurID token.

3. OTP Security Vulnerabilities

OTP-based authentication is one of the most common authentication methods for systems requiring enhanced security. Consequently, methods for unauthorized acquisition of user OTPs, as well as methods for bypassing an OTP authentication step, are highly sought-after in the modern cybercrime community.

This section will present a few of the most common security vulnerabilities and hacking techniques for OTP-based authentication systems. The vulnerabilities and techniques are ordered by their technical complexity, starting from simple cases such as physical loss and theft, to the more complicated malware attacks.

3.1 Physical Loss & Theft

The simplest form of security vulnerability for an OTP-based authentication system is physical loss and theft. Physical theft targets the systems which distribute OTPs using hard copies (section 2.4.1) and physical, proprietary password tokens (section 2.4.4). This is common for many online banking companies, such as the Norwegian BankID. Given a situation where the attacker has successfully attained an username and static password combination, the additional acquisition of a physical password token is enough for a security breach.

Nonetheless, the probability of a large scale theft-based attack is relatively low, due to the time-consuming nature of the attack. Theft of physical devices is easily detected and traced, and the devices can be frozen and rendered useless to the attacker upon detection. As a result, none of the large-scale security breaches in OTP authentication systems are based on physical loss and theft.

3.2 Social Engineering

Social engineering is defined as “attacks carried out on people with the goal of tricking them into divulging some type of sensitive information that can be used by the attacker” (Harris & Maymí, 2016. p. 413). According to the International Criminal Police Organization (Interpol), it is possible to divide social engineering into two categories - **mass frauds** and **targeted frauds** (GlobalSign Blog, 2018).

There is a wide range of techniques used in social engineering, the most common technique being *phishing*, which will be the focus in this sub-chapter. Phishing is a technique to acquire sensitive data, such as bank account numbers, through a fraudulent solicitation in email or on a Web site, in which the perpetrator masquerades as a legitimate business or reputable person

(IETF, 2007, p.222). It is a simple, yet an efficient technique, and is often used to obtain user OTPs in an attempt to bypass two-factor authentication. Phishing can also be used in combination with other hacking techniques, where the user is tricked into downloading malicious software.

3.2.1 Mass frauds

There are different ways of carrying out a phishing attack, however, most mass frauds on end-users of OTP-based authentication systems contain variations of the following steps:

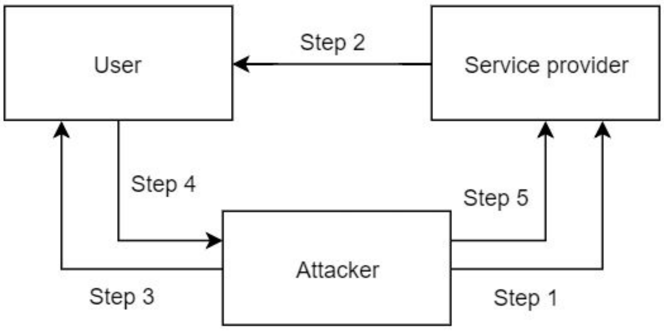


Figure 2 - OTP forwarding attack

1. Attacker requests an action from service provider
2. Service provider generates and sends an OTP to user **[SMS 1 to user]**
3. Attacker tricks user into forwarding the OTP (phishing) **[SMS 2 to user]**
4. User forwards OTP to attacker
5. Attacker completes the request using OTP

(Hossein Siadati, 2017)

In this scenario, the user receives two text messages - one in step two, another in step three, and both can be exploited by the attacker. The attacker can purposefully mimic the language and credentials used in the SMS from the service provider, and generate a similar phishing message. This is most successful in cases where the SMS from the service provider do not contain any warnings, such as Google's two-factor authentication verification message, "your Google verification code is 123456".

In 2016, an experiment was conducted by a team from the University of New York, where the researchers distributed phishing text messages to users of Google, following the steps above, and achieved a 50% success-rate in obtaining the user's OTP. The most successful message, "did you request a password reset for your Gmail account? Delete this message if you did. Otherwise, send 'Cancel' + the verification code we just sent you", was able to achieve 60% success rate amongst the participants (Ibid).

3.2.2 Targeted Fraud

Targeted fraud, in this case, can be interchanged with the term *spear phishing*. Spear phishing is a variation of phishing, where the attack is specifically designed for targeted users. The targeted users or groups often have elevated privileges in the system, or are especially gullible and unaware of security risks.

Spear phishing attacks on OTP-based authentication systems mainly seek to obtain sensitive information about the OTP generation. This can be the encryption algorithm used to generate the OTP code book, or the set of secret keys used to generate the HOTP and TOTP (section 2.3). In addition, the attacker can also obtain the database of password token serial numbers, as well as username and password combinations, to perform large-scale attacks.

RSA's SecurID security breach started as a spear phishing attack, where a small group of targeted, low-profile employees received emails with the title "2011 Recruitment Plan". Once the attachment from the emails were opened, a malicious software was installed and eventually breached the RSA system. Targeted data were stolen, and sent to an external compromised machine (Avivah Litan, 2011). This is a combination of spear-phishing and zero-day malware attack.

3.3 Trial-and-error Password Cracking

One-time passwords generated in OTP-based authentication systems can easily be cracked using various password-cracking techniques, given inadequate security considerations. The simplest form of password cracking is by making repeated login attempts with different passwords until the correct one is guessed (Stuttard & Pinto, 2007. p. 137). This is surprisingly easy for many web- and mobile applications, and can be done by any amateur attacker. In this section, we will first introduce three different types of trial-and-error based password cracking methods. Then, we will present results from our own simple survey about one-time password security strength, and identify specific applications susceptible to these attacks. Results from the experiment indicated that a surprising amount of applications lack even the simplest security measures against OTP cracking attacks.

3.3.1 Brute-force Attack

Brute-force is, by far, the easiest password-cracking method. The mechanism behind this attack is to try every possible combination of characters, until the correct password is guessed. Possible hack steps for a brute-force attack would be the following:

1. Manually submit several bad login attempts for an account you control, monitoring the error messages received

2. After around 10 failed logins and 10+ mins, if the application has not returned any message about account lockout, attempt to login correctly. If this succeeds, there is probably no account lockout policy.
3. Identify a difference in the application's behavior in response to successful and failed logins, which can be used to discriminate between these during the course of the automated attack. (For example URL changes)
4. Create a list of possible usernames (often openly disclosed by the application), or phone numbers from a specific country, and a list of possible passwords (10 000 numbers in the case of a 4-digit code)
5. Use a suitable tool or a custom script to quickly generate login requests using all permutations of these usernames and passwords
(Stuttard & Pinto, 2007. p. 138).

We decided to write a simple python code-snippet, to show the simplicity of these attacks:

```
# This code is based on the snippet from (Teja R D, 2014)
# And modified from Mechanize to Mechanicalsoup so it works in python 3
import itertools
import mechanicalsoup

def brute_force_cracker(login_page_URL, username, number_of_digits):
    combinations = itertools.permutations("0123456789",number_of_digits)
    br = mechanicalsoup.StatefulBrowser()

    br.open(login_page_URL)
    for i in combinations:
        br.select_form(nr = 0)
        br.form["login form"]
        br["username"] = username
        br["password"] = ''.join(i)
        print("Checking ", br.form["password"])
        response = br.submit_selected()

        if response.soup.find("""successful login behavior"""):
            print("Correct password is: "+''.join(i))
            break
```

Listing 1 *- Simple python code snippet for brute-force attacks on HTML forms*

There are also numerous softwares for brute-forcing attacks. Some of the most popular ones are John the Ripper, THC Hydra and Aircrack-ng, each used for different types of password cracking, and can come with premade dictionaries and password hashes. All softwares are readily available for download.

3.3.2 Dictionary Attack

To speed up the the brute-force attack, the attacker can generate a large list of possible passwords, called a “dictionary”. Dictionary attack is defined by the IETF as “An attack that uses a brute-force technique of successively trying all the words in some large, exhaustive list.” (IETF, 2007). A common password dictionary includes a list of frequent passwords, containing words such as “password”, “test123”, “123123” and “starwars”. An example of a free and extensive dictionary is CrackStation’s wordlist, which contains a database of all popular password dictionaries, password leaks and every word in the Wikipedia database (CrackStation, 2018).

3.3.3 Rainbow Table attack

Rainbow table attack is a prime example of a time-memory trade-off in computer sciences, and is used for cracking password hashes (Phillipe Oechslin, u.y). Passwords tend to be stored in databases as *cryptographic hashes*, which cannot be used directly. Rainbow tables use hash functions to precompute a list of passwords hashes from a plain-text dictionary, and stores these as tables, so the time consumed in the cracking process is dramatically shortened. Many rainbow tables are available online for direct use.

3.3.4 A short experiment: Overview and methodologies

Inspired by the brute-force hack steps illustrated in section 3.3.1, we decided to attempt these steps for mobile applications installed on our own mobile devices. We identified a list of applications which used an OTP-based authentication system.

In order for a trial-and-error based password cracking technique to be successful, the application must allow unlimited attempts at guessing the correct OTP. This means the application lack both time-limits (so are *not* time-based synchronous OTPs, see section 2.3.2.2), and limits for the number of login attempts before the the password expires and an account lockout mechanism is activated. Password cracking is also easier when the OTPs generated only contain numbers, and have few digits - a four-digit code can be cracked within 0.1 seconds, using technology from 2015 (BetterBuys, u.y). If the OTP system is, in addition, the only authentication factor in the system, there is a significant risk of security breach, and can easily be cracked using one of the above-mentioned techniques.

For each of the identified applications, we therefore checked the following:

1. Does the OTP expire, or lock out the user, after 10+ failed login attempts?
2. Does the OTP expire after 10 minutes?
3. Does the generated OTP contain more than pure numbers?
4. Is the system a two-factor authentication system?

If the answer is “no” for all four questions, the OTP-based system is classified as highly vulnerable towards a simple brute-force attack.

3.3.5 Presentation of data

Data from the short experiment was sorted into a table (Table 1). Systems significantly vulnerable towards a simple brute-force attack are marked in grey.

Application name	Password expiration after 10+ tries?	Time limit less than 10 min?	OTP more than numbers?	Two-factor authentication?
Digipost	Yes	Yes	Yes	Yes
Tinder	Yes	No	No	Yes
WeChat	Yes	Yes	No	No
Uber	No	Yes	No	Yes
PayPal	Yes	Yes	No	No
Rema1000	Yes	Yes	No	No
Narvesen	Yes	No	No	No
Oslo BySykkel	No	No (but calls after 3 min)	No	No
Espresso House	No	No	No	No
7-Eleven	No	No	No	No
Bik Bok	No	No	No	No
Total Yes	6/11	5/11	1/11	2/11

Table 1 - Safety mechanisms in OTP-based mobile applications

3.3.6 Discussion of results

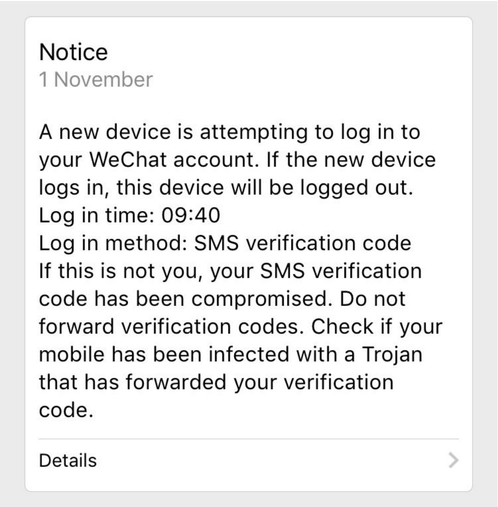
Results from the data collection shows that three out of eleven apps lacked all four security measures, and are therefore susceptible to password cracking attacks. Furthermore, an additional two applications only implemented one of the four security measures. Lastly, WeChat and PayPal, mobile applications allowing users to perform online bank transactions and even functioning as personal IDs, did not use two-factor authentication. Security recommendations for these applications will be discussed in chapter four.

3.4 Man-in-the-Middle (MITM) Attacks

Another common way of illegally acquiring OTPs is through Man-in-the-Middle attacks, also known as MITM attacks. These attacks are defined by the IETF as an “active wiretapping attack in which the attacker intercepts and selectively modifies communicated data to masquerade as one or more of the entities involved in a communication session” (IETF, 2007). There are different variants of MITM attacks, some examples being Man-in-the-Browser (MITB), Man-in-the-PC (MITP) and Man-in-the-Mobile (MITM), and different methods to perform the attack. Typical techniques are SSL and TLS hijacking, where the attacker utilizes vulnerabilities in the insecure HTTP protocol, and DNS (Domain Name System) attacks, where the attacker either forces the user to use a malicious DNS and reroutes all traffic intended for a legitimate server, or introduces corrupt DNS data into the DNS resolver’s cache, causing the server to return an incorrect IP address (Harris & Maymí, 2016, p. 699; Trend Micro, 2017).

It is common to combine a MITM attack with phishing, where the attacker sends a fake phishing email and prompts the user to log into a fake website or application. The attacker presents to the client a fake web-page that looks exactly like the website, and request the user’s credentials. Once this is provided, the attacker can send information back and forth between the user and server, and intercept all traffic in the process (Harris & Maymí, 2016, p. 218).

One of the most known examples of a successful MITM-attack for OTPs is the attack against Citibank in July 2006, where the usernames, static passwords and OTPs were stolen (Rolf Oppliger, 2007). Since then, Citibank has been a victim of multiple, additional attacks (Ilana Greene, 2011).



Notice
1 November

A new device is attempting to log in to your WeChat account. If the new device logs in, this device will be logged out.
Log in time: 09:40
Log in method: SMS verification code
If this is not you, your SMS verification code has been compromised. Do not forward verification codes. Check if your mobile has been infected with a Trojan that has forwarded your verification code.

Details

Figure 3 - WeChat OTP login warning message

3.5 Trojan Horse Attack

OTPs and sensitive information for OTP generation can also be leaked through malicious softwares, often shortened to malwares. Malwares encompasses a wide range of hardware, firmware or software that is intentionally included or inserted in a system for a harmful purpose. Examples of this can be Trojan horse, spyware, virus and worm. In this chapter, we will focus on Trojan horse attacks, as it is the most popular attack for obtaining one-time passwords.

The definition for Trojan horses is “a computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the program" (IETF, 2007).

Trojans can target both smartphones and computers to breach OTP-based systems - in the case of a smartphone, trojans can be designed to intercept text messages containing OTP. The first known piece of malware specifically created for intercepting mTANs (mobile transaction authentication number, a type of OTP) was called ZITMO (Zeus In The Mobile). ZITMO was able to forward the OTPs received in text messages, as well as deleting the SMS so it appears as though no OTPs were ever received (Konrad Rieck, 2013). Since then, a large number of similar softwares have been created. The figure shows an example of a SMS we recently received from the Chinese chatting app, WeChat, containing a warning about mobile OTP trojans.

In addition to forwarding the OTP, it is also possible for trojan malwares to steal sensitive information linked to the OTP generation algorithms, such as the shared secret keys, exemplified by the RSA Securid security breach (Avivah Litan, 2011). According to research in 2013, all known SMS OTP Trojans are user-installed malware. This means the trojan attacks are often performed in combination with a form of social engineering, where the user is tricked into downloading malicious software (Konrad Rieck, 2013).

4. Suggestions to security measures

In this chapter, we will delve into different risk mitigation strategies associated with the above-mentioned threats. The foundation of our suggestions will be the *ISO29115 - Entity Authentication Assurance Framework*, which is an international standard with detailed criterias for the different levels of authentication assurance (AAL), as well as control requirements for each security threats. We will use mobile applications from the short experiment in section 3.3.4 as examples for security improvements.

Additionally, there will also be references to *ISO27001 - Information security management systems Requirements* and *ISO27002 - Code of Practice for Information Security Controls*, as they present a general international standard for information security measurements.

ISO29115 section 10.3.2 suggests an overall control to protect against authentication threats by employing **multi-factor authentication**. This is a requirement for authentication assurance levels 3 and 4. However, multi-factor authentication can reduce the risk and losses from all security threats mentioned in this paper. The applications WeChat and PayPal are therefore recommended to employ two-factor authentication, as security breach in both applications can lead to significant economic loss.

4.1 Protection Against Loss & Theft

The ISO27002 contains code of practices concerning the management of both cryptographic keys and mobile devices used to generate and store sensitive information. Both sections address the risk of physical loss and theft. Additionally, it addresses the control of physical security perimeter, thus minimizing the risk of theft for an organization.

According to the ISO27002, a key management system should be based on an agreed set of standards, procedures and secure methods for issues such as the storage of keys, how authorized users obtain access to the keys, dealing with compromised keys, revoking keys when they should be withdrawn (e.g. when a key is compromised), recovering lost or corrupted keys and destroying keys. A detailed key management policy in the organization will reduce the risks of loss associated with physical loss and theft of OTP password tokens.

Furthermore, the guidance document provides mobile device policies in section 6.2.1, where mobile devices are advised to be physically protected against theft. A specific procedure should also be established for cases of theft or loss of mobile devices. Devices which carry important and sensitive information should not be left unattended and, where possible, should be physically locked away, or special locks should be used to secure the device.

Organizations are also advised to arrange training for the personnel using hardware tokens and mobile devices, in order to raise awareness. Additionally, the organization should protect physical areas which contain either sensitive or critical information and information processing facilities. A detailed implementation guidance for a secure physical perimeter can be found in ISO27002 section 11.1.1.

4.2 Protection Against Phishing

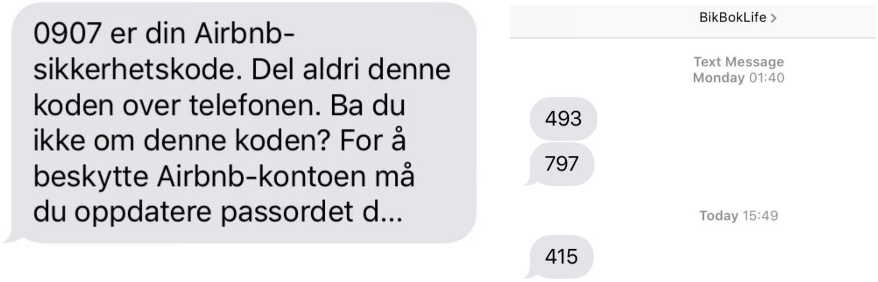
The ISO29115 section 10.3.2 contains three technical control requirements against phishing attacks in information systems, respectively “detect phishing from messages”, “adopt anti-phishing practice” and “mutual authentication”.

“Detect phishing from messages” concerns the implementation of specifically designed controls to detect phishing messages, such as using various spam-filtering techniques (ISO/IEC 29115:2012, 10.3.2.1). Examples of spam-filtering techniques can be Bayesian filters, IP blacklists and URL-based filters. Similar practices can also be extended to voice-phishing (vishing) or SMS-phishing (smishing), by creating spam number blacklists, blocking anonymous numbers and installing caller-ID softwares.

“Adopt anti-phishing practice” is explained as general practices such as disabling images, disabling hyperlinks from untrusted sources, and providing visual cues in email clients. This reduces the risks of users clicking on potentially misleading links and malicious pages (Ibid).

Finally, “mutual authentication” is when the two communicating entities must authenticate to each other before passing data. An authentication server may be required to authenticate to a user’s system before allowing data to flow back and forth (Harris & Maymí, 2016. p.728). This also provides a protection against MITM (Man-In-The-Middle) attacks.

Research from the University of New York concerning the Google OTP phishing experiment (section 3.2.1) concluded that authentication service providers should also add warning messages in the SMS containing OTP (Hossein Siadati, 2017). A simple message such as “if you did not request this password, your account may be compromised (...)” dramatically reduces the risk of users falling prey to phishing attacks. This can be exemplified by the comparison between AirBnb and Bik Bok’s verification code SMS, which we received as a part of the experiment (Figure 4). Organizations sending messages similar to Bik Bok should consider improving the message content.



0907 er din Airbnb-sikkerhetskode. Del aldri denne koden over telefonen. Ba du ikke om denne koden? For å beskytte Airbnb-kontoen må du oppdatere passordet d...

BikBokLife >

Text Message
Monday 01:40

493

797

Today 15:49

415

Figure 4 *- AirBnb versus Bik Bok’s OTP text message.*

In addition to the technical controls, ensuring human resource security is also a crucial defense against phishing attacks. ISO27001 section 7 states that the organization must ensure both the necessary resources, competence, communication and awareness needed for the establishment, implementation, maintenance and continual improvement of the information security management system. Examples of code of practices can be found in ISO27002 section 7 - Human Resource Security, where sub-section 7.2.2 provides a detailed guidance for information security awareness, education and training. This includes for example the personnel’s needs to become familiar with information security policies, standards, laws and regulations, as well as basic information security procedures.

4.3 Protection Against Trial-and-error Password Cracking

ISO29115 uses the term “Offline Guessing” to describe a trial-and-error password cracking method, and suggests four technical controls to protect against these types of attacks. They are “strong password”, “credential lockout”, “default account use” and “audit and analyze”. In this section, we will discuss “strong password”, “credential lockout” and “audit and analyze”, as they are the most relevant for OTP-based authentication systems.

“Strong password” is about enforcing the user to use complicated passwords. An example would be complex, non-dictionary strings that contains a mixture of upper case, lower case, numeric and special characters (ISO/IEC 29115:2012, 10.3.2.1). In the case of an OTP, the authentication system can avoid generating purely numeric characters, and rather use a combination of letters and numbers. Additionally, the OTP generation mechanism must be random enough, so the attacker cannot predict future passwords basing on previously generated passwords.

“Credential lockout” refers to a lockout or slowdown mechanism which must be activated after a certain number of failed attempts (ISO/IEC 29115:2012, 10.3.2.1). The metrics of the mechanism should not be disclosed to the users - for example, informing the user to “try again in X minutes”, or “there are X tries left”. This can allow attackers to optimize the brute-force process, in spite of the policy. In the case of an OTP-authentication system, there should be an additional time-limit mechanism, where the password expires after a certain time period has passed. It is therefore most secure to use a time-based synchronous OTP generation method (section 2.3.2.2). Additionally, it is important to enforce these measures for all authentication challenges within the system - this can be the login itself, but also the functions to change password and recover password (Stuttard & Pinto, 2007. p. 167).

There are several problems linked to existing account lockout mechanisms. First of all, this could allow the attacker to deny service to legitimate users by repeatedly disabling their account, causing a denial of service (DDoS) attack (Stuttard & Pinto, 2007. p. 168). Secondly, account lockout is ineffective against one password to many usernames attack (OWASP, 2017). OTP systems are more resistant against these types of attacks, as instead of account lockouts, the OTP can simply expire and be replaced by a new one after certain triggers.

“Audit and analyze” means that an audit trail of failed logins should be used to analyze for patterns of online password guessing attempts (ISO/IEC 29115:2012, 10.3.2.1). For example, a large amount of failed login attempts from the same IP address can indicate an attacker. An excessive usage and bandwidth consumption from a single use, or failed logins from alphabetically sequential usernames or passwords can also indicate a brute-force attack (OWASP, 2017).

Our short experiment in section 3.3.5 identified three mobile applications - Espresso House, Bik Bok and 7-Eleven, which appeared to lack mechanisms for account lockouts and password expiries following frequent failed attempts. We would greatly advice these applications to review their security settings, especially in the case of Espresso House, where users can save their bank credentials and enable one-click purchases. Furthermore, ten out of eleven mobile applications’ OTPs contained purely numbers, with Bik Bok’s OTP being only 3 digits. This poses increased security risks to the user and the system.

4.4 Protection Against Man-in-the-Middle attack

There is an extensive list of measures against MITM attacks, depending on the technique employed by the attacker. As OTP-based authentication systems are vulnerable to MITM attacks in the same manner as other systems, the general protection recommendations against MITM attacks can be extended to OTP-based systems. The ISO29115 lists two measures against MITM attacks - firstly the aforementioned *mutual authentication* (section 4.2), and secondly having an “encrypted session” (ISO/IEC 29115:2012, 10.3.2.1). An encrypted session means that all data shared between the client and the server should be encrypted. Should the data be compromised or stolen in the process, it cannot be decrypted and read by an attacker.

It is encouraged to use HTTPS (Hypertext Transfer Protocol Secure), which is an internet communication protocol that “protects the integrity and confidentiality of data between the user’s computer and the site”. Data sent with HTTPS is secured via TLS (Transport Layer Security protocol), and provides encryption, data integrity and authentication (Google, 2018).

Other methods of protection against MITM is by using an NIDS (*Network Intrusion Detection System)* or VPN (*Virtual Private Network)* to secure the network. NIDS uses sensors, analyzers and contain administrator interfaces. The sensor detects activity traffic, which is sent to the analyzer. Once abnormal traffic is detected using various detection techniques, a warning is sent to the administrator (Harris & Maymí, 2016. p.822). Examples of popular NIDS are Snort and Bro (Audun Jøsang, 2018).

Unlike NIDS, VPN uses encryption and tunneling protocols to ensure security (Harris & Maymí, 2016. p.649). It is a private and secure connection through an untrusted network, and offers protection even in cases when the user is forced to use a non-HTTPS site or while using a malicious wifi access point.

Finally, it is possible to protect against MITM attacks by using a browser that supports HSTS (HTTP Strict Transport Security), which is a security mechanism that allows servers to request browsers to only interact with HTTPS, instead of using the insecure HTTP. This prevents TLS stripping attacks, a type of MITM attack (Audun Jøsang, 2018).

4.5 Protection Against Trojan Horse Attacks

ISO27001 section A.12.2 requires organizations to ensure controls against malwares, by implementing measures to detect, prevent and recover from malware attacks, combined with user awareness. A list of guidance against malware protection is presented in the parallel section in ISO27002.

Some general advices from ISO27002 is to create blacklists and whitelists of websites and applications, thus preventing and detecting uses of unauthorized software. Reviews of data supporting critical business processes, updates for malware detection softwares and personnel training should be conducted on a regular basis. Using two or more software products protecting against malwares from different vendors can increase the effectivity of the softwares. Email attachments, downloads and other files received over networks should be scanned before storage and use. The scan must occur multiple times, at different stages of the file transfer.

In the specific case of OTP-authentication systems, it can be helpful to add a warning about potential risks of malwares in the OTP messages, such as in the case of WeChat (Figure 3). Additionally, as Trojan horses are mostly user-installed, all protection measures against phishing attacks will simultaneously protect against Trojan attacks.

5. Conclusion

This paper has presented an in-depth analysis of the mechanisms behind OTPs, common security threats and risk mitigation strategies. It shows how a large-scale attack against companies such as RSA is possible, and how an OTP-based system is vulnerable to common attacks.

All in all, an OTP authentication system should be based on an information security standard, such as the ISO-series. The safest OTP authentication system appears to be a combination of time-based synchronous OTP generation method, multi-factor authentication system using different channels, a secure network, and deliverance through a hardware device stored in a secured physical perimeter. Additionally, all users and administrators of the system must be trained and be thoroughly aware of the security risks.

Our research indicates that protecting an OTP-based authentication system against common attacks is by no means a trivial task. The system must be thoroughly planned, continuously improved and maintained, to counter the increasing complexity of potential attacks. Many organizations lack the necessary security measures to protect themselves and their users against a security breach.

Finally, our own experiments identify at least three popular mobile applications which appear to be vulnerable to simple brute-force attacks. We have discussed multiple attack strategies, and potential safeguards against these. Whilst the protection against complex attacks is difficult, basic security features such as not allowing the user to attempt the OTP more than a certain number of times in a timeframe, improving the OTP text messages or making the OTP too complex to brute force are relatively straightforward to implement. This constitutes a significant security risk for both the organization and their users, and it is therefore of paramount importance to improve the way many organization decide to implement OTPs.

6. Bibliography

Alcodes (2016). *One time password SMS and its use case for different industries* [Internet]. Available from: <https://medium.com/alcodes/one-time-password-sms-its-use-case-for-different-industries-63b8e5465bdf > [Read 25.10.2018].

Angela Moscaritolo (2011). *RSA confirms Lockheed hack linked to SecurID branch* [Internett]. Unknown location. Available from: <https://www.scmagazine.com/home/security-news/rsa-confirms-lockheed-hack-linked-to-securid-breach/> [Read 20.10.2018].

Art Coviello (u.y). *Open Letter to RSA Customers* [Internet]. Unknown location. Available from: <https://www.sec.gov/Archives/edgar/data/790070/000119312511070159/dex991.hm > [Read 21.10.2018].

Audun Jøsang (2018). *Lecture 01* [Internet]. Oslo. Available from: <https://www.uio.no/studier/emner/matnat/ifi/IN2120/h18/lectures/in2120-2018-h01-intro-basics.pdf>

Audun Jøsang (2018). *Lecture 9: User Authentication* [Internet]. Oslo. Available from: <https://www.uio.no/studier/emner/matnat/ifi/IN2120/h18/lectures/in2120-2018-h09-user-authentication.pdf> [Read 16.10.2018].

Audun Jøsang (2018. *Lecture 11: Network Perimeter Security* [Internet] Oslo. Available from: <https://www.uio.no/studier/emner/matnat/ifi/IN2120/h18/lectures/in2120-2018-l11-netperisec.pdf>

Avivah Litan (2011). *RSA SecurID attack details unveiled - lessons learned.* [Internet]. Unknown location. Available from: <https://blogs.gartner.com/avivah-litan/2011/04/01/rsa-securid-attack-details-unveiled-they-should-have-known-better/> [Read 22.10.2018].

BankID (u.y). *Hverdagsmagi. Garantert papirfritt* [Internet]. Oslo. Available from: <https://www.bankid.no/bedrift/bruksomrader/> [Read 21.10.2018].

Barkan, Elad; Biham, Eli; Keller, Nathan (2006). *Instant Ciphertext-Only Cyptanalyses of GSM Encrypted Communication** [Academic paper]. Israel Institute of Technology. Available from: <Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication by Barkan and Biham of Technion (Full Version)> [Read 30.10.2018].

BetterBuys (u.y). *Estimating Password-Cracking times* [Internet]. Unknown location. Available from: <https://www.betterbuys.com/estimating-password-cracking-times/> [Read 23.10.2018].

CrackStation (2018). *CrackStation's Password Cracking Dictionary* [Internet]. Unknown location. Available from: <https://crackstation.net/buy-crackstation-wordlist-password-cracking-dictionary.htm> [Read 21.10.2018]

Dafydd Stuttard & Marcus Pinto (2007). *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*. 1st edition. Indianapolis: Wiley Publishing Inc.

Datatilsynet (u.y). *Når er det krav om sterkere autentisering (for eksempel to-faktor) enn bare brukernavn og passord?* [Internet]. Oslo. Available from: <https://www.datatilsynet.no/regelverk-og-verktoy/verktoy/sporsmal-svar/Informasjonssikkerhet-hos-virksomheter/nar-er-det-krav-om-sterkere-autentisering/ > [Read 24.10.2018].

Deepak Venkatesh (2016). *The Story of OTP* [Internet]. Unknown location. Available from: <https://gomedici.com/the-story-of-otp/> [Read 25.10.2018].

Eric Chabrow (2011). *RSA Breach Costs Parent EMC $66.3 Million* [Internet]. Unknown location. Available from: <http://www.govinfosecurity.com/articles.php?art_id=3913> [Read 19.10.2018].

Federal Financial Institutions Examination Council (u.y). *Authentication in an Internet Banking environment* [Internet]. Unknown location. Available from: <https://www.ffiec.gov/pdf/authentication_guidance.pdf > [Read 26.10.2018].

GlobalSign Blog (2018). *What is Social Engineering? The Human Confidence Game* [Internet]. Unknown location. Available from: <https://www.globalsign.com/en/blog/what-is-social-engineering-the-human-confidence-game/> [Read 20.10.2018].

Google Play (2017). *Google Authenticator* [Internet]. California. <https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=en> [Read 17.10.2018].

Google (2018). *Secure your site with HTTPS* [Internet]. Unknown location. Available from: <https://support.google.com/webmasters/answer/6073543?hl=en> [Read 02.10.2018].

Ilana Greene (2011). *Citigroup Data Breach: A Lesson and Warning For All* [Internet]. Unknown location. Available from: <https://www.forbes.com/sites/ilanagreene/2011/06/13/citigroup-data-breach-a-lesson-and-warning-for-all/> [Read 22.10.2018].

ISO/IEC FDIS 27001:2013(E) (2013). *Information technology — Security techniques — Information security management systems — Requirements.* International Organization for Standardization, Geneva, Switzerland. Available from: <https://wiki.uio.no/mn/ifi/IN2120-2018/img_auth.php/5/5d/ISO27001-2013.pdf> [Read 13.10.2018].

ISO/IEC FDIS 27002:2013(E) (2013). *Information technology — Security techniques — Code of practice for information security controls*. International Organization for Standardization, Geneva, Switzerland. Available from: <https://wiki.uio.no/mn/ifi/IN2120-2018/img_auth.php/0/09/ISO27002-2013.pdf> [Read 13.10.2018].

ISO/IEC FDIS 29115:2012(E) (2012). *Information technology — Security techniques — Entity authentication assurance framework*. International Organization for Standardization, Geneva, Switzerland. Available from: <https://wiki.uio.no/mn/ifi/IN2120-2018/img_auth.php/5/55/ISO29115-2013.pdf> [Read 13.10.2018].

Mulinner, Collin; Borgaonkar, Ravishankar; Stewin Patrick (u.y). *SMS-Based One-Time Passwords: Attacks and Defense** [Academic paper]. Technische Universit¨at Berlin. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.678.114&rep=rep1&type=pdf> [Read 13.10.2018].

IETF (Internet Engineering Task Force) (2005). *HOTP: An HMAC-Based One-Time Password Algorithm* [Internet]. Unknown location. Available from: <https://tools.ietf.org/html/rfc4226> [Read 30.10.2018].

IETF (Internet Engineering Task Force) (2011). *TOTP: Time-Based One-Time Password Algorithm* [Internet]. Unknown location. Available from: <https://tools.ietf.org/html/rfc6238> [Read 16.10.2018].

IETF (Internet Engineering Task Force) (2007). *Internet Security Glossary, Version 2* [Internet]. Unknown location. Available from: <https://tools.ietf.org/html/rfc4949> [Read 17.10.2018].

Oppliger, Rolf; Hauser, Ralf; Basin David (2007). *Protecting Ecommerce Against The Man-in-the-Middle* [Internet]. Unknown location. Available from: <http://webtorials.com/main/resource/papers/BCR/paper116/01opplinger.pdf> [Read 18.10.2018].

OWASP (2017). *Blocking Brute Force Attacks* [Internet]. Unknown location. Available from: <https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks> [Read 01.10.2018].

Phillipe Oechslin (u.y). *Making a Faster Cryptanalytic Time-Memory Trade-Off* [Academic paper]. Ecole Polytechnique F´ed´erale de Lausanne. Available from: <https://lasec.epfl.ch/pub/lasec/doc/Oech03.pdf> [Read 16.10.2018].

Regjeringen (2008). *Rammeverk for autentisering og uavviselighet i elektronisk kommunikasjon med og i offentlig sektor* [Internet]. Oslo. Available from: <https://www.regjeringen.no/no/dokumenter/rammeverk-for-autentisering-og-uavviseli/id505958/> [Read 28.10.2018].

Rieck, Konrad; Stewin, Patrick; Seifert, Jean-Pierre (2013). *Detection of Intrusions and Malware, and Vulnerability Assessment*. 1st edition. Berlin: Springer. Available from: <https://link-springer-com.ezproxy.uio.no/content/pdf/10.1007%2F978-3-642-39235-1.pdf> [Read 17.10.2018].

Sahir Hidayatullah (2010). *Man in the middle attack prevention strategies* [Internet]. Unknown location. Available from: <https://www.computerweekly.com/tip/Man-in-the-middle-attack-prevention-strategies> [Read 16.10.2018].

Siadati, Hossein; Nguyen, Toan; Gupta, Payas; Jakobsson, Markus; Memon, Nassir (2017). *Mind your SMSes: Mitigating social engineering in second factor authentication* [Academic paper]. Unknown location. Available from: <https://reader.elsevier.com/reader/sd/pii/S016740481630116X?token=1F09FB1A5B92139DF777A78475D1568F3E7179D6C8D8834412F9D940FAE26A3486CA7BF5AF703CF58ABA1A9D8DFEF002> [Read 22.10.2018].

Siemens (2007). *Two Factor Authentication* [Internet]. United Kingdom. Available from: <https://web.archive.org/web/20120112172841/http://www.insight.co.uk/files/whitepapers/Two-factor%20authentication%20(White%20paper).pdf> [Read 18.10.2018].

Shon Harris & Fernando Maymí (2016). *CISSP Exam Guide*. 7th edition. New York: McGraw-Hill Education.

Standard Norge (2017). *ISO-Standarder* [Internet]. Oslo. Available from: <https://www.standard.no/standardisering/iso-standarder/> [Read 01.11.2018].

Teja R D (2014). *Brute Force A Website Login In Python* [Internet]. Unknown location. Available from: <https://coderinaero.wordpress.com/2014/12/08/brute-force-a-website-login-in-python/> [Read 30.10.2018].

Trend Micro (2017). *Infosec Guide: Defending Against Man-In-The-Middle Attacks* [Internet]. Unknown location. Available from: <https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/infosec-guide-defending-against-man-in-the-middle-attacks> [Read 03.11.2018].

UNIX and Internet Security (u.y). *6.4 Common Cryptographic Algorithms* [Internet]. Turkey. Available from: <http://web.deu.edu.tr/doc/oreily/networking/puis/ch06_04.htm#PUIS-CHP-6-SECT-4.7> [Read 22.10.2018].

Webroot (u.y). *What is Social Engineering? Examples and Prevention Tips* [Internet]. Unknown location. Available from: <https://www.webroot.com/ie/en/resources/tips-articles/what-is-social-engineering> [Read 24.10.2018].

6.1 Companies mentioned in this paper

7-Eleven	<http://www.7-eleven.no/?>
BankID	<https://www.bankid.no/bedrift/>
BikBok	<https://bikbok.com/no/>
Digipost	<https://www.digipost.no/>
Espresso House	<https://no.espressohouse.com/>
Lockheed Martin Cooporation	<https://www.lockheedmartin.com/en-us/index.html>
Narvesen	<https://www.narvesen.no/>
Oslo BySykkel	<https://oslobysykkel.no/>
PayPal	<https://www.paypal.com/no/home>
Rema1000	<https://www.rema.no/>
Tinder	<https://tinder.com/?lang=nb>
Uber	<https://www.uber.com/no/nb/>
WeChat	<https://www.wechat.com/en/>
Yubico	<https://www.yubico.com/>