UNIVERSITY OF BERGEN

DEPARTMENT OF INFORMATICS

Experimental Study on One-Time Passwords used in Authentication within Norwegian Banking

Author: Sivanja Naguleswaran

Supervisor: Igor A. Semaev

June, 2020

Acknowledgments

First and foremost, I would like to thank my supervisor, Igor A. Semaev. I am thankful for his help and guidance throughout this thesis.

I would also like to thank my fellow master students at the Department of Informatics and everyone at the Selmer Center, who has made this an enjoyable learning experience.

A huge thanks to my family and friends for their constant support, despite having no clue what my degree is about. Finally, I would have to thank my parents for their unconditional faith and love in me. A special thanks to my little sister for her encouraging words and acts of caring, especially during the Corona-pandemic and self-isolation.

Abstract

The use of dynamic one-time passwords plays a significant role within online banking in Norway. This study examines four tokens provided by DNB, Sparebanken Møre, Sparebank 1 and Nordea used in token-based authentication and the one-time passwords they generate. By studying one-time passwords collected at various time intervals it was able to reconstruct the internal token-algorithm and the verification protocol. This research argues that three out of four tokens indicate weaknesses that can have damaging effects. This is also proven by explaining a basic theoretical attack, which demonstrates that the success probability of an attack is higher than the expected probability of 10^{-6} or 10^{-8}.

Table of Contents

CHAPTER 4

CHAPTER 5

APPENDIX A

List of Acronyms and Abbreviations

PIN	Personal Identification Number
ID	Identification
OTP	One-Time Password
FRR	False Reject Rate
FAR	False Acceptance Rate
2FA	Two-Factor Authentication
MFA	Multi-Factor Authentication
PKI	Public Key Infrastructure
HMAC	Hash-based Message Authentication Code
DES	Digital Encryption Standard
TDES	Triple Digital Encryption Standard
AES	Advanced Encryption Standard
TOTP	Time-based One-Time Password
HOTP	HMAC-based One-Time Password
OATH	Initiative for Open Authentication
OCRA	OATH Challenge-Response Algorithm
SBM	Sparebanken Møre

Chapter 1
Introduction

1.1 Background

Authentication is a vital part of this fast-growing, digitalized world. Fundamentally, today's society is more reliant on computer technology and digitalization than ever before. The cases of identity thefts, phishing, and cyber-attacks have increased alarmingly within the last two decades, leaving the need for more reliable and secure authentication methods [1]. The last couple of years have also shown the value of sensitive data for attackers, and several companies have taken a stand against cybercrimes and upped their game on information integrity and security.

Today, different authentication processes can be used to validate a user's identity. Each process provides various levels of security. One of them being the Two-factor authentication based on a token generated one-time password. Especially in Norway, these are often used within online banking and public services. The public mostly uses the authentication scheme for day-to-day activities, such as paying bills online or accessing an official account, but it is also used frequently to access enterprises in a wide range of sectors. This means that the authentication scheme must be highly secure with a small probability for a computer breach. This thesis will examine how the token generated one-time passwords works, if the combinations are predictable and if the expected security standards for an attack are upheld.

Note that the term **security or cyber security** is applied to a variety of contexts. In this thesis the word security is used when referring to methods or products that require protection from malicious attacks. The attacks can be everything from cyberattacks, data breaches, theft, etc., where the main goal is to attempt to modify, destroy, obtain information without authorization. These kinds of attacks are also collectively referred to as **basic attacks**.

1.2 Motivation

As token-based authentication is a scheme that is widely used by individuals within all age groups, it is often easy to forget the damaging effects if one’s credentials are stolen. With the rise in cloud services, smartphones, and the Internet of Things, a person can store dozens of online accounts online. With just one click, a user can access both usernames and passwords from different devices, and from experience it is known that devices and accounts can be easily hacked by a third party. This makes a dynamic one-time password, such as a password generated by a token, more reliable security-wise. Since tokens produce random digit passwords, it is more difficult for an attacker to predict or hack compared to a static password, which often remains the same after each round of authentication. Or that is the expectation at least.

The results from an experimental study conducted on DIGIPASS GO3, a token provided by Sparebanken Vest, demonstrated that the patterns are more predictable than expected [2]. This aroused the curiosity around the other banks in Norway and the reliability of the tokens they provide. The primary source of motivation was experimenting and testing devices that thousands of people use every day around Norway. Finding results not matching our expectations was merely a bonus.

1.3 Summary of Results

This study highlights a couple of weaknesses found in token generated one-time passwords provided by DNB, Sparebanken Møre, Sparebank 1 and Nordea. Three out of four tokens indicate that the leftmost digit is predictable and used for synchronization. In tokens provided by Sparebanken Møre and Nordea, it is possible to see that the leftmost digit either increases by a + 1 mod 10 or remains the same as previously generated password. With the token provided by Sparebank 1, the first digit of every generated password combination is 9.

Tokens from Sparebanken Møre and Nordea shows that the last five- and six digits of every combination are not distributed uniformly. By analyzing the collected data, it appears that the numbers 0,1,2,3,4,5 are almost used twice as much than the remaining digits 6,7,8,9.

Based on the experiments it attempted to reconstruct the internal algorithm of each token, along with the verification protocol. The reconstructed algorithm seems to apply for the collected data and explains repetition patterns. With the accumulated data, a theoretical attack is presented. The attack proves that the tokens from Sparebanken Møre, Sparebank 1, and Nordea do not uphold the optimal probability. The success of an attack is $8^{-5}, 10^{-5}$and 8^{-6} for the respective banks, instead of the expected probability of 10^{-6} and 10^{-8}.

1.3 Structure of the Thesis

This thesis is organized into the following chapters:

Chapter 2: Authentication and Online Banking in Norway

This chapter describes what authentication is and different types of authentication. The various authentication methods used in online banking in Norway, as well as the functionality of these methods, will also be explained.

Chapter 3: One-Time Passwords

In this chapter, one-time passwords and the process in which they provide an additional security layer in authentication will be explored. The published algorithms that provide one-time passwords will be discussed along with some of the companies that offer such services in the form of tokens.

Chapter 4: Experimental Study

This is the core of the thesis, where the experimental study and analysis are presented. The chapter starts with a brief explanation of the research this study is based on. Furthermore, there will be a detailed presentation of the setup for the experiment and the approach, followed by a breakdown analysis of each bank.

Chapter 5: Discussion and Conclusion

The chapter contains a summary of the findings and the effects on a larger scale. This will be followed by recommendations for further work and a conclusion.

Appendix A

All the collected data in the form of tables are categorized according to chapter and banks. The data is presented as tables with different timesteps in an increasing order.

A1: Password combinations collected with DIGIPASS GO3 – *provided by DNB.*

A2*:* Password combinations collected with DIGIPASS GO3 – *provided by Sparebanken Møre.*

A3: Password combinations collected with Gemalto Lava Token – *provided by Sparebank 1 Nordvest.*

A4: Password combinations collected with HID Pocket Token – *provided by Nordea.*

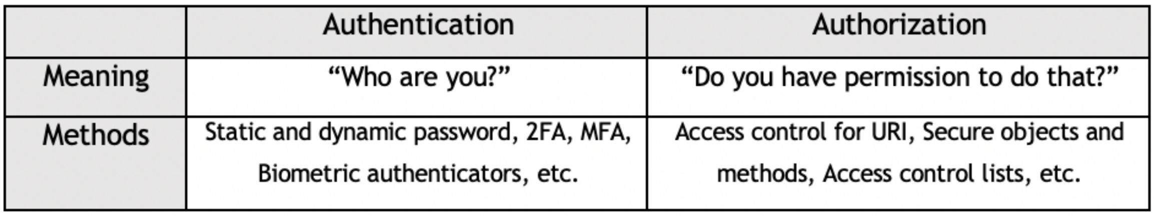
Chapter 2

Authentication and Online Banking in Norway

2.1 What is authentication?

«Colloquial uses of the term 'authentication' are occasionally misleading. For example, authentication is neither authorization nor identification» [3, p. 33]. When an individual is logging in to a system, they are required to prove that they are who they claim to be. To verify their identity, a user often provides an identifier, such as an email address. They do this in combination with a password that is associated with the address to get authenticated. By definition, this means identity is the idea of describing an entity uniquely. Therefore, authentication is defined as the process of identifying an individual, such as a user of a computer system, and establishing confidence accessing a secure domain.

Authorization is another term that is often linked with authentication. Authorization determines which resources, meaning rights, actions, and privileges the person has access to based on his or her verified identity [4]. These concepts are distinctly different and often distinguished from each other in the world of cyber security. Again, using the previous email login example, by using an email ID and password, one gets authenticated and can get authorized into an email account to the specific mail-address, but not permitted access to anyone else's account.



	Authentication	Authorization
Meaning	"Who are you?"	"Do you have permission to do that?"
Methods	Static and dynamic password, 2FA, MFA, Biometric authenticators, etc.	Access control for URI, Secure objects and methods, Access control lists, etc.

Table 2.1: A clearer picture of what differentiates authentication form authorization.

There are many methods of authentication for an individual. However, the choice of authentication depends on the identifier. An identifier can be anything that points to the entity which is being identified and then authenticated—for instance a name, social security number, mail ID, or telephone number. Identifiers are categorized as strong and weak based on the uniqueness of the pointer.

For example, the social security numbers that are provided for each individual in Norway are a unique mapping for a specific person within the country, thus categorizes as a strong identifier. A phone number, on the other hand, can be classified as a weak identifier since it in many cases can be applied and linked to several individuals. However, a combination of multiple weak identifiers can produce a strong identifier and possibly be seen as a unique identification of an entity [3, p. 42]. A prime example to understand this concept is a telemarketer who tries to contact potential customers. A marketer identifies the individual by his or her number. The marketer can easily reach them by looking up the information stored on that specific individual in their database, such as name and surname combined with address and telephone number. The individual, however, cannot reach a particular marketer by just having a number. This is because a marketer usually uses a standard company number where one often (not always) gets redirected to a switchboard. This makes it harder to find the specific marketer.

2.2 Authentication factors

Authentication is not a simple concept, but it is the first step of accessing control. The process is required across several online platforms, from accessing social media to online banking. Having determined that authentication is essential, and it is possible to achieve in several ways, the next important step is to find a suitable method for achieving individual authentication. Commonly, the most methods used for authentication are based on three factors: something you know, something you have and something you are [5, p. 88].

2.2.1 The knowledge factor: Something you know

The most common factor used in authentication is something personal that only the user knows. This can include credentials associated with the individual that consist of information that only the user understands and possesses. One of the most used knowledge factors in the world is passwords.

A personal password is often user created. It can merely be a numeral code, a random string, or a combination of both string and numbers. It is also easy to memorize and recall for the particular user, which makes it more convenient and more time-efficient. Some systems also use cognitive passwords, which traditionally exists of multiple, and often personal questions. Such questions would and should only be known by the user.

Static passwords and personal identification numbers (PIN) is a common factor used daily worldwide, but it is also the most uncomplicated technique to beat. Anything one can easily memorize and recall often ends up being the authentication factor an individual needs to get validated, making it far too easy for attackers to hack [6, p. 14]. A knowledge factor is also natural to share and can become highly insecure during an extended period. Another case is when a user often uses the same static password for authentication of different systems. The password can then become very predictable.

2.2.2 The possession factor: Something you have

A possession factor is a technique that includes physical objects that one owns as an identification factor [6, p. 11]. These kinds of objects can be identifications papers or cards (ID), electronic ID cards, ATM cards, hand-held tokens, and devices for digital signatures. Identification papers and cards traditionally contain a picture with the name, date of birth combined with personal security number and a signature. Documents can often be more detailed than cards, but are still effective. Electronic ID cards contains a chip embedded in the card, which stores personal information about the user.

When a user needs to authenticate themself in person, it is common to use ID paper or cards, such as passports and driver's licenses. To an individual, this is the easiest accessible method and a more time-efficient option for authentication. Such objects of identification are often valid for an extended period, which makes it more reasonable to carry around and use locally and internationally. Therefore, these kinds of documents have to be highly secure, meaning it should not be possible to alter, copy, or modify the information the identification contains. For this very reason, smart cards are used together with a knowledge factor, for example a PIN-code. This method is called a two-factor or multi-factor authentication. This concept is addressed later in chapter 2.3 Types of Authentication Methods.

Tokens are often associated with gaining access to a bank account online. When logging in, the system will ask for a one-time password (OTP) from a token. Token, a knowledge factor, gives users access to their bank account. More details about OTP and tokens will come later in Chapter 3.

2.2.3 Inherence factor: Something you are

Simply explained, an inherence factor is something the specific individual has. This type of authentication works by observing an individual user's physical characteristics based on biometrics. Biometric characteristics can be everything from a fingerprint, signature, biometrical samples, scanners to face recognition, patterns and lines in the iris and voice analysis [7].

Biometric authentication is an accurate, quick, and user-friendly tool explicitly designed to create an entirely reliable and secure authentication system. Today, biometric authentication methods are widely used, implemented, and built into several daily used devices, like phones and computers. The first fingerprint sensor phone was already released in 2011 by Motorola Atrix [8]. Many smartphones in the digital market has today both fingerprint and face recognition software. This type of biometric authentication helps unlock a phone and authorizes downloads, purchases and login to different applications.

Since biometric authentication schemes base their authentication on information that is already stored, it continually has to match inputted information from an individual to existing information in the database. This factor makes it hard to expect an exact match every time and is therefore susceptible to errors [9, p. 26]. It is common to divide the types of error into two groups: Type 1 error and Type 2 error. Type 1 errors – also known as false positives – occurs when a system fails to recognize a user that is already registered and known to the system. Type 2 errors or false negatives is when the system acknowledges an unknown user as known and fails to reject the user into the system. Combining Type 1 and Type 2 errors gives us this following error matrix:

Actual Value

		True	**False**
Measured or Perceived value	**True**	Correct True positive	Type 1 error False positive
	False	Type 2 error False negative	Correct True negative

Table 2.2: A confusion matrix that represents an error that can take place when using biometric authentication.

The sensitivity of the environment where biometric authentication is in use dominates the False Reject Rate (FRR) – the probability of a system will fail to reject an imposter, and False Acceptance Rate (FAR) – the likelihood that a system will fail to authenticate a known user into the system [9, p. 27]. If one chooses to use biometric authentication in a regular workspace, the implementation of the authentication will allow some Type 2 errors. If the system does not allow any Type 2 errors, it is stringent and needs to have a perfect match between the biometric data and the data in the stored database. In the case of a system that does not allow any Type 2 errors, the possibility of getting Type 1 errors are very high, which means the users of the system are not going to be able to get authenticated every time.

To create an optimal and secure system, there needs to be a balance between Type 1 and Type 2 errors. To tackle this problem, all systems which use biometric authentication have some parameters that can be tuned and adjusted to find a point where FAR and FFR are equal. When the rates of the two are equal, the typical value is called Equal Error Rate or Crossover Error Rate [10, p. 293].

Finding a balance is the key to a well-functioning system, but it is also crucial that each environment evaluate functionality versus security before determining the balance. In some settings, for example within the government which consists of highly confidential information, Type 2 errors may not be permitted. As a consequence, they must then accept that the users may have to authenticate themselves several times. In other words, one calibrates the rate of the environment depending on the level of protection that it needs.

2.3 Types of authentication methods

Since technology and digitalization has developed well, the importance of security has also evolved. Today's environment demands more secure and reliable authentication schemes that offer more protection. The types of authentication methods are divided into three main categories.

Single Factor Authentication is the oldest and most basic form of authentication [11, p. 214]. This method, also known as primary authentication, requires an identifier, such as a name, number or a mail address, and an authentication factor to authenticate a user. In most cases, the authentication factor used is a knowledge factor, which, in combination with an identifier, gets the user access to his/her account.

Even though this method is used widely on several platforms, experience and history have shown that static passwords are quite easy to access and are vulnerable to hacking. While combining different elements to create a secret password does minimize the risk of brute force, passwords are still associated with poor security standards and can easily be stolen.

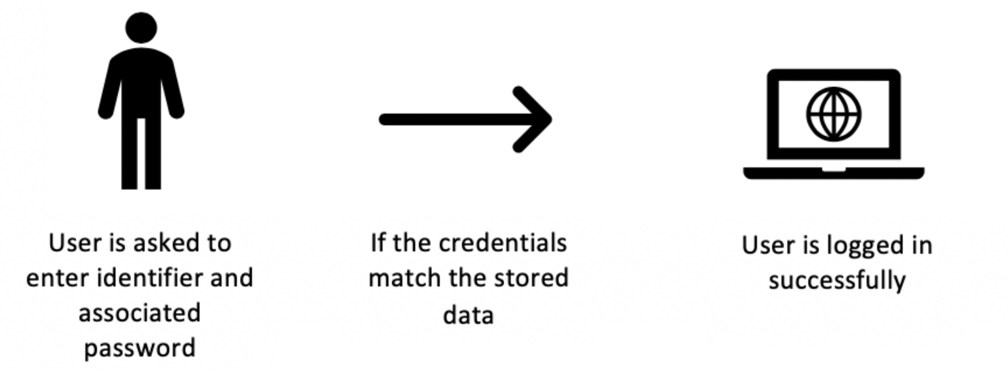

User is asked to enter identifier and associated password
If the credentials match the stored data
User is logged in successfully

Figure 1: Shows the necessary steps of a Single-factor authentication.

Two-Factor Authentication (2FA) is an authentication method that takes a step further than primary authentication and creates a significantly more secure and robust solution. Combined with an identifier and an authentication factor, this method requires an additional factor to be able to authenticate a user into a system. By adding another layer of complexity, this makes the authentication protocol more reliable and secure to verify a user's identity.

In fact, according to a security threat report published by Verizon Data Breach Investigations last year [1, p. 19] reused and weak credentials like passwords cause 80 percent of data breaches related to cyber security. The report also highlights that such data breaches could easily be prevented by a system that requires 2FA authentication. Increasing the number of security layers from one to two gives immediate results where the attacker now has two different layers to crack. This additional layer also makes the probability of getting access to an unauthorized account much smaller than having only one factor (Figure 2).

Two-factor authentication systems are also well implemented in the real world to add extra security. If an individual wants to use an ATM-machine, it is necessary to have either a credit or a debit card (a possession factor), in addition to a personal PIN-number (a knowledge factor) in order to get authenticated. Having one without the other will not be sufficient to gain access. The same principle takes place in computer security systems. The most used combination when it comes to authentication within computer systems using 2FA is using something the user knows, such as a password or a personal PIN, and generate a one-time password using a token or a smartphone, which is in the user's possession.

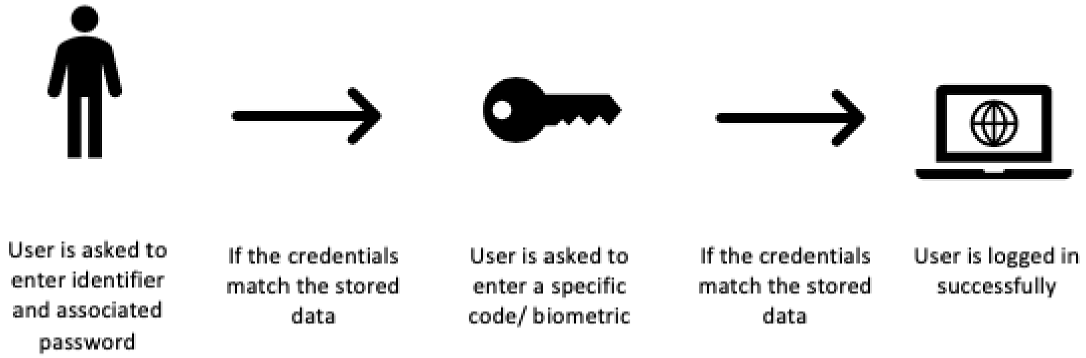

User is asked to enter identifier and associated password
If the credentials match the stored data
User is asked to enter a specific code/ biometric
If the credentials match the stored data
User is logged in successfully

Figure 2: Shows the necessary steps of a Two-Factor Authentication.

Multi-Factor Authentication (MFA) is, as the name suggests, an authentication method that uses any two or more factors. The technique requires that the factors used must be different from each other to grant access to a system. This means one cannot apply two layers from the same factor in order to strengthen the authentication. For example, using an authentication method consisting of a static password and a PIN-number will not get categorized as MFA since both factors belong to the category knowledge factor.

2FA is a subset of MFA. The only difference is that 2FA always has two different factors, but MFA can have two or more factors to authenticate user's identity. In robustness, MFA gives a more secure layering and it is more reliable when it comes to authentication so that the correct user accesses the system. It has already been disclosed that passwords are weak and can be altered, hacked, and stolen. This weakness also applies to physical objects like cards, tokens, and ID-papers. By adding multiple layers, this strengthens the security and makes it harder to brute force or hack into a system. The probability that an attacker could obtain both user ID and the same user's password to that exact phone addition to stealing the user's fingerprint is quite low. MFA gives the extra security layer some system requires. Inherence is also one of the most challenging factors to steal, which makes it quite valuable. It makes the authentication more secure and less predictable for attackers to steal. The chance is still there, but quite minimal.

With this being said, several systems still use 2FA, or even single-factor authentication, as a method to identify an individual. The main reason behind this choice of method is because it is easier for the end-user and much more cost-efficient, which is the main priority for most system developers. Many users do not appreciate if an authentication solution is too slow, unreliable, or complicated. This aspect has in many cases led companies to favor cheaper and cost-efficient methods over more secure options, but in recent times this trend has changed. Recent attacks on big organizations show how valuable and sensitive data is to hackers and third-party intruders [12]. Organizations such as banks and hospitals are trusted with incredibly sensitive information, which is vital to protect. These environments have to protect crucial and highly sensitive user data like health, financial and identity-related information. The risk of having security breaches in such an environment will cause damage on a large scale and which increases the necessity of having more reliable and secure authentication.

2.4 Authentication in Norwegian Banks

All Norwegian banks offer authentication services through the internet in addition to many other services. When users need to access their accounts, they need to authenticate themselves to receive access. This chapter will look into how authentication in Norway works, and which methods Norwegian banks offer for customer authentication.

2.4.1 Norwegian National Identity Number

The first step of authentication is to have an identifier that points to an identity. The standard identifier for every customer in Norwegian banks is a national identity number issued by the Norwegian Tax administration (In Norwegian: Skatteetaten). In Norway, every citizen on the Norwegian National Registry has an identification number called "fødselsnummer" or directly translated "birth number." According to the Norwegian Tax Administration, an identification number is issued to everyone born and settled in Norway, citizens born Norwegian or residents abroad who need a national identity number to get a Norwegian passport [13].

The identity number consists of 11 digits where each number alone or combined with another represents unique information of the individual. The 11 digits are composed of two parts, where the first six digits represent an individual's birthdate in the following format:

$$DDMMYY$$

The second part is called a personal number (in Norwegian: personnummer) and consists of five digits. The personal number is again a combination of two parts, where the first three out of five digits represent *individual numbers*, and the last two digits represent the *control numbers*. This combination gives us the following format:

$$I_1I_2I_3C_1C_2$$

The individual numbers $I_1I_2I_3$ are assigned sequentially for everyone with one specific birthday. The last digit of the identification number I_3 also represents the gender of the given individual, where an even number represents female and the odd number represents male. The control digits C_1C_2 is a checksum composed of the previous nine digits in the identification number.

The Norwegian Tax Administration also issues something called a D-number. This number gets assigned to individuals who do not fulfill the requirements of getting an ordinary Norwegian national identity number but still need an identification to use in Norway [14]. The D-number also consists of 11 digits and is composed of two parts. The first part is a modified version of the date of birth written in the same format. An individual's date of birth gets amended by adding "4" to the first number. For example, if the date of birth is the 6th of March 1997, the first digits of the D-number will show 460397. The control digits also have the same format and are calculated after modifying the date of birth.

2.4.2 ID-porten

Norwegian banks offer several ways for a customer to authenticate themselves and access their accounts digitally. The majority of the methods provided by the banks are presented by ID-porten, which is a digital login-system developed and managed by the Norwegian Digitalization Agency (In Norwegian: Digitaliseringsdirektoratet or Digir). This system is a standard login system in Norway. It makes it possible for all users with a Norwegian national identity number to access more than 1000 public services run by Norwegian government agencies [15].

It is important to note that ID-porten is not a separate login platform itself but provides different login solutions through several methods. Currently, ID-porten offers five different approaches to achieve electronic identification with varying levels of security between levels of 3 and 4. While level 3 gives the user access to most official digital platforms, it will not grant access to personal accounts that contain sensitive information. For example, to gain access to the official web-portal of medical care, Helse Norge, requires the highest level of security [16]. In case of a user is already logged in and tries to access a service with a higher level of security, the system will automatically let the user know the issue and what to do next.

MinID is an electronic ID solution of security level 3, which is equivalent to medium-high. It is also the only solution of this level, thus a recommended as a method of authentication to be used in most of the public services. The method uses 2FA authentication, where the user has to enter their Norwegian national identity number as an identifier and 2FA consisting of a static password, a knowledge factor, and a PIN code, a possession factor. The PIN code can be sent either on SMS to the user's phone or from a PIN code letter produced and sent by the government directly to the user. To be able to receive OTPs through SMS, one needs to use the PIN code letter to login and register the phone number.
Input:

1. Norwegian national identity number
2. Static password
3. OTP sent by SMS or PIN code from PIN code letter

BankID is the solution that the majority of the Norwegian population use daily, both as a digital ID and online signature. Today, it has reached around four million registered users in the country. The solution was under development in 2000 and was soon after made available to Norwegian consumers in 2004 [17]. BankID provides the highest level of security that ID-porten has to offer. It is used not only for all government-related services (for example: tax payment and healthcare services), but also by several independent enterprises within all sectors such as Norwegian banks, which also requires robust authentication methods. BankID is based on Public Key Infrastructure (PKI), a framework that manages certificates and keys, and is required to provide asymmetric cryptography and digital signature services. Thus, it supports authentication for accessing accounts, making online payments, signing public and personal documents online, placing bids for real estate, applying for loans, and registering documents publicly.

The solution uses the same 2FA authentication method as in MinID. In order to get authenticated, a user still needs a Norwegian national identity number and a personal static password in addition to a dynamic password generated by an OTP-device. What differentiates MinID from BankID is that BankID has a higher level of security and uses a token-generated dynamic password in addition to a static password. The token can only be obtained by contacting the bank and by having a meeting with them in person. A user does not need to repeat this process multiple times because BankID is a standard method used across all banks in Norway. Once an OTP-device is activated and received by the user, it is possible to order a new device just by logging in with the "old" one.

There are two different methods of BankID: normal BankID and BankID on mobile phones. All banks in Norway offer both services, and an individual can have access to both methods at the same time.

Normal BankID

For accessing normal BankID, one needs to have an OTP-device. This device can be either a physical token provided by the bank or a downloadable app accessible on a smartphone.

Input:

1. Norwegian national identity number
2. OTP generated by the security token
3. Personal static password

BankID on mobile phones

For accessing BankID on a mobile device for the first time, one needs to login to the bank with a normal BankID and activate BankID on mobile.

Input:

1. Phone number and date of birth (DDMMYY)
2. Personal PIN-number on the phone (four to eight-digits)

Between step 1 and step 2, a reference code will appear on the screen. As a verification, the same code is sent to the user's phone and will have to be verified. If the code is identical to the one appearing on the screen, the user clicks "Accept".

Buypass ID is a solution delivered by a Norwegian IT-firm. The Norwegian Digitalization Agency states the following: "Buypass ID is delivered on the smart card with electronic ID on different levels. One can order or upgrade an existing qualified electronic ID at the highest security level (level 4)" [18]. To order any of the services Buypass provides, one needs to contact them directly through their website.

Buypass ID on a smart card

Buypass requires a card reader in addition to having a smart card with a Buypass ID. A smart card reader can be bought as a device and plugged into a device of choice.

Input:

1. Smart card
2. PIN-code

Buypass ID on mobile

The authentication protocol is different depending on if the consumer is using Buypass ID on an app or have a Buypass password. Either way, these inputs are necessary to get authenticated through Buypass ID on mobile.

Input:

1. Buypass ID
2. Norwegian national identity number
3. Instructions on phone

Commfides provides the same high-security level as BankID and BuyPass ID. Management of Commfides is a Norwegian company and provides authentication based on PKI. The solution delivers smart cards, like Buypass, but also on USB drives with smart cards. To order Commfides, one needs to contact them directly or pay a visit to their headquarters in Lysaker.

When using components regarding Commfides authentication solutions for the first time, Java installation is required in order for it to work.

Input:

1. Commfides eID
2. Personal PIN-code

Chapter 3

One Time Passwords

As technology and priority of making every user interaction as digital as possible have drastically grown, the need for secure and reliable user authentication and the exchange of sensitive information through and across computer systems has become more and more critical. The previous chapter described several methods for authentication, but one technique that has stood the test of time is passwords. It is a well-known solution, and users often make their own static passwords to make it personal and easy to remember.

A user usually chooses a static password and often remains the same from its creation until the user decides to change the password by modifying it for the specific account. Even though it is a standard authentication scheme, it is only secure to a certain extent. Passwords are easily a target of phishing and getting eavesdropped by a third party. This weakness often makes it easy for an attacker to gain access to the system without any authorization. The downside of having a static password well-known to the user is in many scenarios solved by having a dynamic password instead or combined with a personal password creating an additional layer of security based on 2FA.

3.1 What is an OTP?

A one-time password, often referred to as OTP, is a password combination that one only can use once. It creates a dynamic numeric code, where each digit is randomly chosen and thus creates a unique password each time. This password is often used in an authentication session as a one-time password for an individual user of a system and then thrown away [19, p. 885]. Because it is only valid for one session of authentication, it is more impervious for a lot of attacks.

After one round of authentication, the password is no longer valid, thereby useless, and the user needs to request a new one-time password for the next session. Thus, if an attacker or potential intruder captures an OTP of many others over the network, it still has no use since it has already been used for an authorized authentication process or to process a transaction that is no longer valid. Another main advantage of this type of password is that the digit-combination is highly unlikely to be hacked. To determine the next possible password based on the current one is extremely hard and highly unlikely. In addition to this, many one-time passwords are time-based, meaning they have a specified timeframe. The given password must be used as a valid password for authentication of an individual, or else it gets expired. This requirement makes it harder for criminals to get their hands on the password in time to use it within the time interval ends.

Since an OTP is not something the user knows and can memorize, it is produced by something a user has, a possession factor. Physical objects, such as phones and tokens, are typical devices that generate OTP combinations used for authentication [20, p. 134].

Phones are one of the most used devices for accessing OTPs. The OTP is delivered through text messages as a ubiquitous property and is a reliable communication channel. The equipment being mobile is also a significant benefactor, where it can be brought everywhere and be accessible to everyone at any time. Text messaging on phones often has text-to-speech and more responsive design to make it easier for everyone to read, even if it is a numerical password. In recent times several banks have developed apps designed to produce OTPs, which are all downloadable on all smartphones. A user must get authorized online enable to gain access to the app [21].

A token is a keychain-like object that only has the sole purpose of producing OTPs. By pushing the button, the token generates and displays a numerical password. The generated password is usually in-between a six to ten-digit number depending on the manufacturer or provider of the token. Since it is a one-time password, the digit displays for a certain amount of time before it disappears, ensuring it is not accessible to everyone that is authorized to use the same code again.

3.2 Cryptography

There exists a lot of ways to generate OTPs based on different algorithms as Event-based OTP [22], Time-based OTP [23] and OATH Challenge-Response algorithm [24]. All of these algorithms have been adopted as Internet Engineering Task Force standard RFC 6238 and are published and available to everyone who wants to study them. Even though all of these algorithms fundamentally deliver the same result and outcome, small factors and additional elements distinguish the algorithms from each other.

3.2.1 Definitions

It is necessary to introduce some cryptographic notions and definitions to understand the context of different methodologies and terms used in the other parts of this chapter.

Symmetric cryptography is a process there both encryption and decryption use the same secret key. **Asymmetric cryptography** requires two separate keys, one for encryption and one for decryption. The public key is used for encryption and secret key for decryption.

Block cipher is a symmetric cryptographic algorithm that encrypts/decrypts the input on a fixed size of data. The algorithm uses the same secret key for both encryption and decryption.

Digital Encryption Standard (DES) is a block cipher with a fixed block size of 64-bit and a key size of 56-bit. Due to the small key size, it is considered weak and "broken".

Triple Digital Encryption Standard (TDES) is a variation of DES. The algorithm applies DES three times (encrypt- decrypt -encrypt) to each data block.

Advanced Encryption Standard (AES) is a block cipher of a fixed length of 128 bits and allows three different key lengths of size 128, 192, 256 bits. The algorithm is based on Substitution-Permutation network.

Hash-based Message Authentication Code (HMAC) is a message authentication code that uses a cryptographic hash function and a secret cryptographic key.

Message Authentication Code (MAC) is an encrypted checksum that results from sending data through a message authentication algorithm.

RSA is an Asymmetric cryptography algorithm and is based on a unique property of the prime numbers for encryption.

3.2.2 Cryptographic Hash function

It is necessary to have a basic grasp of what a hash function is to understand how the published algorithms work. "A cryptographic hash function is used to compress a message of arbitrary length to a short, random-looking, fixed-length message digest" [25, p. 8]. Meaning a hash function takes an input string of any length and output a fixed length. Since the function is hard to reverse once it is used to derive a value, it is called a one-way function. Thus, the feature provides data integrity of some settings [26, p. 137].

Hash-based message authentication code (HMAC)

Since the published algorithms are an extension of each other, they are all based on a hash-based message authentication code, often referred to as HMAC. It is important to remember that the published algorithms are not a hash-function, but uses hash functions as a part of their algorithm. It is possible to combine HMAC with any cryptographic hash method. In our case, all of them, by default, are based as SHA1 [22] in combination with a secret key [27, p. 124]. A minor change to the message creates a significant hash difference. HMAC bases its strengths on the properties of the cryptographic hash function that is used. HMAC uses two parameters of hash functions, one for the message input and one for the secret key that is only known for the receiver and sender.

The algorithm of HMAC is defined as [28]:

$$HMAC[K, m] = H[K \oplus 0 \times\ 5C5C \ldots\ \| \ H[K \oplus 0 \times\ 3636 \ldots\ \| \ m]]\ )$$

Where K represents a secret key, H is a hash function (For example, SHA1, SHA2), $0 \times\ 5C5C$ and $0 \times\ 3636$ represent block-sized padding consisting of repeated values of $0 \times\ 5C5C$ and $0 \times\ 3636$ bytes respectively.

3.2.3 HMAC-based One-Time Passwords

This algorithm uses keyed-hashing message authentication code (HMAC) and is often referred to as Event-Based One-Time Password or Hash-based One-time Password (HOTP). This method relies on two-part of information, where the one is a secret key, and the other one is the moving factor, in this case, a counter. The algorithm is event-based, meaning it is based on the increasing value of the counter and a secret symmetric key. The static key is only known to the validation service and token. To create an HOTP-value the HmacSHA-1 hash, which is part of the algorithm, is used. This message authentication code is the one-time password that the user is going to use in the authentication process to get access to the system. Each repetition of the algorithm uses a different hash than the previous one.

The definition of the HOTP algorithm that generates OTP value [22] :

1) Create an HMAC hash from a secret key and counter.

$$Value = HOTP(secretKey, counter)$$

2) Since the user cannot understand or enter a raw output of the 160 bits calculation of the HmacSHA-1 value, the output is truncated into an HTOP value represented by digits. One can write the HOTP function as:

$$HOTP(SecretKey, counter) = Truncate(HOTP_H(secretKey, counter)$$

Where H represents a cryptographic hash function.

Even though the report [22] states that shorter length values of HOTP are more convenient than longer values, the shorter values are more vulnerable to brute-force attacks. The report [22] address this problem by recommending two steps as a possible solution. Step one is to set a throttling parameter T, where the parameter T defines the maximum number of attempts possible for a login-session for OTP validations. Step two is to implement a delay mechanism, where each failed attempt should increase the wait. Thereafter create an additional delay before there is any chance of authorized re-entry for authentication.

3.2.4 Time-based One-Time Passwords

This OTP is based on HOTP and works as a variant and further development of the algorithm described in the previous section. In Time-Based One-Time password, also often referred to as TOTP, the moving factor is replaced by time [23]. So instead of using the counter for deriving the passcode, TOTP uses time and a stored secret key as inputs to calculate the OTP.

TOTP uses the time-factor to make time intervals or timesteps, which means the password is only valid a specific period. The interval between each unique password is traditionally between 30 to 90 seconds. When the time limit expires, the password is outdated, and one has to generate a new password for authentication of an individual to get validated by the system.

The algorithm remains the same except the counter factor is replaced with time, denoted by T, as a moving factor. There is no definition of a hash function in this algorithm, even though SHA-1 often is used in practice where it generates a hash value of 160-bit. Therefore, the formulation for a TOTP algorithm is [23]:

1) Create an HMAC hash from a secret key and current time in Unix time.

$$Value = HOTP(secretKey, current_T)$$

Where $current_T$ is a counter that keeps track of completed timesteps between the initial time (referred to T_0) and the current time.

2) Since the time value changes every second, the generated passcode is only valid for a particular time and has to create another one. The required formula is:

$$current_T = floor((unixT(now) - unixT(T_0))/T_1)$$

Where
Floor is a function used to round the value down to the highest integer less than or equal to.
unixT(now) is Current time in Unix
unixT(T_0) is Unix time at T_0
T_1 represent the timestep in seconds, which the generated OTP will be valid for

3.2.5 Challenge-Response Algorithm

Challenge-Response Algorithm is an algorithm developed by the Initiative for Open Authentication (OATH). The algorithm, which is often referred to as OATH Challenge-Response Algorithm (OCRA), is a modified version of HOTP. Unlike time or event-based authentication, this algorithm uses different parameters to generate an OTP rather than just a secret key and an incremented counter or a limited time interval [24].

Challenge and Response is a security mechanism based on communication between the verifier and the person being authenticated. The verifier will communicate by asking a question, a *challenge* question, to the prover which has to provide a valid *response* to get verified and move on to the next process [24].

The algorithm requires a cryptographic function that is performing the computation of OCRA, in addition to the secret key K and input parameters, which defines this algorithm as:

$$Value = CryptoF(SecretKey, Input_D)$$

Where
$CryptoF$ is defined as a function that performs the OCRA computation based on secret key and data input.

The input parameters defined by $Input_D$ represent an array of operations of the OCRASuite value. “An OCRASuite value is a text string that captures one mode of operation for OCRA, completely specifying the various options for that computation” [24]. Thus, $Input_D$ is defined as:

$$Input_D = \{OCRA_s | \, 00 \, | \, \mathrm{C} \, | \, \mathrm{Q} \, | \, \mathrm{P} \, | \, \mathrm{S} \, | \, \mathrm{T}\}$$

Where

$OCRA_s$ is a value representing the operation to compute an OCRA response

00 represent separator value in byte form

C is an unsigned counter value of 8 bytes, where the high-order bits get processed first. The value must be synchronized between all parties to work.

Q is a concatenated list of challenge question(s). The list must be 128 bytes. The Q should be padded with zeros to the right if the value is less than 128 bytes.

P is a hash value of PIN or passwords. This secret code is known to all the parties.

S is a string that holds the information of the current session. The string is UTF-8 encoded and defined by the OCRASuite value.

T is timesteps in Unix time. The value represents an unsigned integer of 8-bytes.

All the parameters mentioned above are optional, except parameter Q $(\mathrm{Challange\ Question})$ which is mandatory.

The concatenation order for a response is always:

$$Response = \{OCRA_s | \, \mathrm{C} \, | \, \mathrm{P} \, | \, \mathrm{S} \, | \, \mathrm{T}\}$$

Where C is the verifier generated challenge question and P | S | T represents the prover generated response/question.

The algorithm offers both one-way and mutual Challenge-Response of authentication, besides electronic signature possibilities.

During a **One-way Challenge-Response,** the verifier will question a randomly generated challenge value to the prover. The prover must then use the challenge in the computation described above, eventually send the response back to the verifier.

Mutual Challenge-Response takes the process a step further where both servers and clients have to authenticate each other mutually. The client sends a random challenge to the server. The server responds with a computed response along with a server-challenge. The client has to check if the server computed response is correct and then calculate a client response that is sent back to the server. The authentication protocol is completed when the server verifies the client's response to the challenge.

3.3 OTP Providers

In the past two decades, the growth of digital authentication and the need for secure and reliable communication has increased. Parallel to this, the demand for digital companies that specialize in digital security and manufactures safe and reliable devices, has grown exponentially. This growth has made the market for digital appliances and services more attractive for companies to focus on and provide for the customers, which often is even more prominent companies/organizations in need of a digital service and a specific device.

This chapter will look further into some of the companies that provide such services in the form of OTP. Note that the manufactures picked are of the relevance to this study and the primary analysis, and have nothing to do on a larger scale. All of the companies mentioned below have a direct connection to the banks picked in this research and all are using their services. They all provide authentication of some sort. Since this study also mainly focuses on hard tokens, there will be a section about what type of tokens each company offers.

3.3.1 Gemalto

Gemalto is one of the international companies that produce products for digital authentication. The company, established in Amsterdam, is the result of merging two companies, Axalto and Gemplus Internationals [29]. Their headquarters are still in Amsterdam, Netherlands, and are today acquired by Thales to get a head start in the fast-growing market of this highly digital world. Gemalto specializes in digital security, where they provide software applications and personal devices. The company mainly focuses on tools that are used in authentication, such as hard-tokens, smart cards, and managed services. They are also the largest manufacturer company of SIM cards.

As mentioned, hard tokens are one of the products Gemalto provides, consisting of more advanced transaction verification and signing methods with different algorithms, including the Challenge-Response Algorithm [30]. The design prioritizes customers with no further experience with digital tokes. The company manufactures several types of tokens. All of them have different types of authentication methods, and thus differs in security level and design flexibility.

Gemalto Flex BLE: A thin and flexible token that provides strong authentication with a dynamic signature.

Gemalto Signer Token: Risk-based token specially developed to mitigate the most attacks. This also provides authentication with a dynamic signature.

Gemalto Pico token: An ultra-thin security token that is compliant with OTP-algorithm OCRA.

Gemalto Lava token: A lightweight security token consisting of just one button, which generates and displays the OTP on the screen of the token. The OTP is generated by either TOTP or HOTP-algorithm. This model is also the type of token that is used by most of the banks in Norway, such in Sparebank 1 Nordvest.

3.3.2 OneSpan

OneSpan was founded under the name VASCO Corp. in 1984 and today a highly listed company that specializes in digital identity and solutions to prevent fraud in the digital world [31]. Those solutions include everything from identity verification of an individual to risk-based anti-fraud adaptive authentication. In their own words, "Over 10,000 customers, including more than half of the world's top 100 banks, trust OneSpan to secure their digital journeys" [32].

The company is most known for their solutions within multifactor authentication, especially DIGIPASS technology, a solution for generating OTPs for efficient and secure two-factor authentication. The prototyping and early development of DIGIPASS and generating OTPs started in 2000 and was marketed internationally as a comprehensive solution for verifying an individual of a user into a system [33]. The authentication method was released nine years after development, both as tokens and applications for App Store, made just available for technology supported by Apple products [34].

OneSpan has today three versions of DIGIPASS available, where each of them provides the same type of authentication which generates the OTPs through hard tokens [34]. All of them are mainly used as a solution to enhance the security for banks.

DIGIPASS GO 3: This is the token DNB and Sparebanken Møre provides to its customers in Norway for online authentication. The token supports two types of crypto algorithms, DES and TDES and can be used in MFA combined with a knowledge factor, such as a static PIN-code.

DIGIPASS GO 6: This one is more advanced than the previous token since it supports the AES crypto algorithm in addition to DES and TDES. The token also supports HOTP and TOTP so that it can be adjusted and changed according to the costumer's wishes.

DIGIPASS GO 7: Supports the same algorithms as DIGIPASS GO 3 and DIGIPASS GO 6 but is the only token with a security level of FIPS 140-2 of Level 2. This feature means it has upheld the security standards made by the Federal systems for shielding sensitive information within a system [32].

3.3.3 HID Global

With customers in over 100 countries, HID Global is a company that produces devices that makes authentication and verification of an individual easier and more convenient [35]. HID Global was founded in 1991 and works today as an independent company under the brand Assa Abloy. The American company, which has its headquarters in Austin, Texas, was ideally founded to develop and produce identification devices based on radio technologies. HID Global primarily manufactures tools that help with identity verification and security products helping people to navigate through digital networks without any complications.

When it comes to authentication through OTP, they offer several tokens, which all generate dynamic passwords to provide users with simple and easy access into the systems [36]. All of them must be used combined with a knowledge factor, which in most cases is a static password, in addition to the generated password. Today HID Global offers seven different OTP tokens, which all provide multifactor authentication. Unlike the other tokens disclosed in this thesis, each of these tokes looks different design-wise and provides various services to match the specific target groups.

OTP Mini Token: This is made especially for custom orders, where customizing the token in favor of the customer is an option. The token is designed especially for organizations where employees and consumers need to get authenticated daily. It supports both time- and event-based authentication. When using this token, one has to validate the PIN on a server to be able to get authenticated.

OTP Flexi Token: A pin-pad token that one also can customize with personal details. In addition to being high on functionality and quite flexible, it is also one of the most cost-efficient tokens HID Global produce. It supports both time- and event-based authentication.

OTP Pocket Token: This token is designed especially for mobile users. This element makes the token portable and is easy to access and use anywhere. This model is also the token Nordea provides for its customers in Norway. The token features a keypad, where one has to type a PIN to unlock the device before it generates an OTP which the user can use, indicating a challenge-response method. It supports both time- and event-based authentication, as well as OATH HOTP and OATH TOTP.

OTP One Token: A similar token to the previous one but rather designed for suppliers, employees, partners, and contractors than everyday users. It supports both time- and event-based authentication, as well as OATH HOTP and OATH TOTP. It needs a PIN validation on the device before generating the OTP.

OTP Keychain Token: A small and compact token that is easy to carry and designed for suppliers, employees, partners and contractors. It supports both time- and event-based authentication, as well as OATH HOTP and OATH TOTP. This also needs a PIN validation on the device before generating the OTP.

OTP Desktop Token: A token with a larger display and buttons than the traditional tokens, which makes it ideal for elderly users, employees and office-based users. For extra safety precautions, the token has a simple PIN-code. One has to enter the code before generating an OPT, which indicates a challenge-response method. It supports both time- and event-based authentication, as well as OATH HOTP and OATH TOTP.

BlueTrust Token: One-click authentication with frictionless access- The user can authenticate an individual into systems and data networks based on Bluetooth and NFC [37].

Chapter 4
Experimental Study

The chapters above defined what an OTP is and why it is needed in most 2FA and MFA authentication methods. It has also been looked into different manufacturers and the types of tokens they produce. Hence, it is now time to look at the core of the study. As explained in Chapter 2, several of the authentication methods used in different Norwegian banks require token generated OTPs. The way that the passwords are generated will be further discussed in this chapter. Since experiments conducted in this thesis are based on an experimental study [2], there will be a brief description of that paper and the results. Subsequently, there will be a presentation of the setup for the experiment and the approach before diving into the analysis of each bank and presenting the results.

4.1 Starting point

The inspiration behind this thesis is a study written by Igor A. Semaev [2]. The study describes an experiment based on DIGIPASS GO3, a token provided by Sparebanken Vest. In his research, he describes how the six-digit password combination is produced by DIGIPASS GO3 and proves that the password has a constant pattern which is more predictable than expected. Additional details about the OTP's synchronization function and algorithm are produced and presented. The study also argues that the probability of a successful attack is 8^{-5}, which is much higher than the expected probability of 10^{-6}. The research highlights the success probability if one or many customers are targeted and how this can affect the security and integrity over an extended period.

Based on Semaev's [2] study on DIGIPASS GO3, together with the growing need for a secure and reliable authentication, the idea of applying the experiment to other tokens was formed. This chapter describes a study based on four different tokens provided by various banks established in Norway.

4.2 Experimental Setup

All banks in Norway offer authentication either through ID-porten or one of their own designs. This sub-chapter will present the data collected by token-based authentication, where a possession factor generates the OTP. Note that although ID-porten provides other authentication methods that require OTPs, for instance MinID and BankID on mobile, this analysis focuses just on BankID and how the algorithm works.

4.2.1 Approach

The experiment on token-based OTP is based on different banks established in Norway: Sparebanken Møre, DNB, Sparebank 1 and Nordea. While describing how authentication in Norwegian banks works in chapter 2, it was made clear that to gain authorization to log into an account through BankID, customers have to get in contact directly with the specific bank. After receiving the token, the customer has to go through a particular protocol to make the authentication work. This protocol differs from bank to bank, but after being authorized to use BankID and token as an OTP generator, the procedure is fundamentally the same.

The easiest way to approach this problem is to open up an account and ask to get authenticated through BankID and get a token provided by the bank. In Norway, it is also possible to use one BankID provided by one bank to access another bank system with the same BankID. Realistically, this means a customer only needs one token to generate an OTP and can access different accounts provided by various banks in Norway. Even though this element did not affect the experiment, it is necessary to mention that an ideal customer does not need to have four different tokens to access four different banks. Therefore, receiving several tokens was purely done for the experiment.

Each bank has different distributors providing the OTP. OneSpan manufactures tokens both for Sparebanken Møre and DNB, Gemalto for Sparebank 1, and HID Global for Nordea. The manufacturer of the tokens had no impact while collecting data since the banks can choose the algorithm and modify specific details internally within the organization. This concept is also proven when analyzing Sparebanken Møre and DNB, since they both have the same manufacturer.

4.2.2 Data collecting

In order to do the analysis, a large number of passwords generated from different tokens were collected. The data presented in this thesis was collected by generating passwords and observing the time intervals with a stopwatch. The intervals have been carefully considered and structured according to each bank and the token algorithm. Note that the given value of time in the tables is approximate to the real-time of pressing and generating the OTP. Fluctuations are minimal since the method and process have been repeated several times during this experiment. It can also easily be repeated and validated by others. All data presented in this thesis is collected independently with no input from the respective banks other than providing the OTP-tokens as a general account holder of a bank. As the experiment and the data collected is time-based, the intervals can either be in a three-digit or four-digit form. This means if the interval between each generated token is 53 seconds, then it will be presented as 0:53 and five minutes as 05:00. Digits before the colon represent minutes and after seconds.

Since the experiment required four different analyses for the various banks, the collected data will be presented separately and categorized according to each bank. Even though the procedure and approach for collecting data were the same, it is easier to see the patterns of each bank and compare them. This allows for a clearer perspective of each bank and the choices made relative to the algorithm used in the OTP token. Therefore, each section in the following chapter will contain a detailed description of how the data was collected and justify the chosen time intervals. Thereafter, an analysis based on patterns of the collected data, as well as the verifier's part of the authentication, will be presented.

4.2.3 Analysis

The analysis contains a five-part study of the collected data:

1) Analyzing possible patterns and try to create a **general algorithm**. The algorithm should apply as a standard formula for the collected data.

2) An overview of the **digit distribution** of the collected data and analysis of the digits used in generating a password.

3) **Reconstructing the algorithm** of how to generate an OTP. As mentioned, there are only three publicly published algorithms that are available to study, so the assumption was that all provided tokens from each bank are based on these. By studying the patterns, it is possible to get an idea of the internal construction of an OTP.

4) In most cases, token generated OTPs are used when a user is authenticating themselves into a system. This system can be either on a computer or on a cell phone. Either way, the server has to verify the generated OTP and to do so, it has some **verification protocols** in place. Therefore, the thought was to reconstruct a verification protocol and check how it handles delays.

5) The analysis is concluded with a **possible basic attack** where the static password is already known to the attacker. The tokens in this thesis produce passwords containing either six or eight

digits. Therefore, the optimal probability is expected to be 10^{-6} or 10^{-8} depending if all ten possible numbers are evenly distributed.

4.3 DNB

With more than 2.1 million retail customers, DNB is the largest financial service group in Norway [38]. With branches all over Norway and 19 different countries around the world, DNB offers everything from financial services to real estate and insurance. It is safe to say with more than 1.5 million active users of Internet banking in Norway, authentication and security procedures around user validation has to be top priority in order to maintain a secure system [38].

4.3.1 Authentication

DNB offers three methods of authentication: BankID, BankID for mobile phones, and Without BankID (In Norwegian: BankID på mobil and Uten BankID). Both BankID and Without BankID requires a token. Without BankID requires that the user has to input a personal four-digit PIN code in addition to an OTP generated by the provided token. If a customer wants to use Without BankID as an authentication method, they cannot use a token that is provided by another bank, unlike BankID.

OneSpan

To generate OTPs, DNB uses a token called DIGIPASS GO3. The OTPs generated in this experiment are provided by a token with serial number 37-0234699-5 (Figure 3). The token is manufactured by VASCO, as OneSpan was once named.

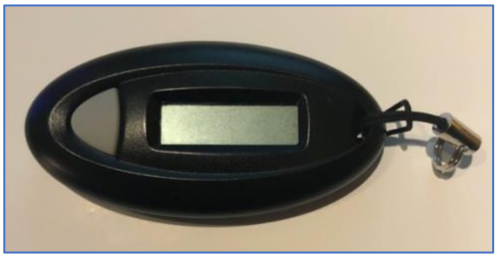


Figure 3: *A picture of token model DIGIPASS GO3 provided by DNB*

To generate an OTP, the button on the token must be pushed. Followingly a number consisting of six digits will appear on the LCD-display. The combination will be displayed on the screen for exactly 30 seconds before disappearing. When one tries to regenerate a password 30 seconds after the initial press, a new OTP will appear on the display.

4.3.2: Data collecting

OTPs generated between time steps of (0:15+,0:30+)

The experiment was conducted at different time intervals to determine possible repetition between the generated OTPs. In total, 516 password combinations were generated with this particular token. Since the password combination stays on display for 30 seconds, the experiment started by generating OTPs with an interval of 30 seconds and the process was repeated 50 times (Table 4.3.1).

1	427467	1
2	434511	1
3	752615	1
4	795468	1
5	041795	1
6	607251	1
7	091658	1
8	785859	1
9	790210	1
10	142903	1
11	483306	1
12	251893	1
13	094396	1
14	135750	1
15	883954	1
16	109630	1
17	524835	1
18	768962	1
19	141570	1
20	154005	1
21	946984	1
22	334123	1
23	982834	1
24	735101	1
25	631690	1
26	256523	1
27	641081	1
28	877597	1
29	163243	1
30	264309	1

Table 4.3.1: *An extract of Table 4.3.10 6-digit combination at time interval 0:30+. The full table can be found in Appendix A.*

As the table above states, there is no sign of repetition of a password at an interval of 30 seconds. The first combination was produced at 0 seconds and pressed 30 seconds after the first OTP was generated. As mentioned, the timestamp displayed in the table is approximate since it was all timed and pressed manually using a stopwatch.

Even though the OTP stays on display for 30 seconds, it is still possible to press the button twice, one to make the combination disappear and another for regenerating another password. Since there is a chance of repeating passwords, the existing time interval was reduced by 10 seconds. By reducing the previous time step of 30 seconds to 20 seconds between each press, it was tried to generate a password for every 0:20+ (Table 4.3.2). It is possible to notice repeating combinations. The interval was reduced further by doing the same experiment of 15 seconds between two consecutive OTPs. The repetition appears for every other password combination with a repetition pattern 1 – 0 – 1 – 0, where one represents the generated new password and 0 represents the repetition of a previous password (Table 4.3.7 in Appendix A).

When seeing repeated sequences and password combinations, it was concluded that the threshold for DIGIPASS GO3 token generating a new password and repeating the previous one happened between 0:20+ and 0:30+. To determine the exact timestamp where the token always generates a new unique password, an attempt was made to generate a six-digit password with the time interval of 29 seconds containing 50 passwords (Table 4.3.9).

1	336448	1
2	329342	1
3	329342	0
4	835663	1
5	997727	1
6	997727	0
7	529944	1
8	338560	1
9	338560	0
10	868048	1
11	868048	0
12	269993	1
13	824128	1
14	824128	0
15	204037	1
16	908755	1
17	908755	0
18	611599	1
19	611599	0
20	008277	1
21	485641	1
22	485641	0
23	651281	1
24	324823	1
25	324823	0
26	841999	1
27	448676	1
28	448676	0
29	964969	1
30	964969	0

Table 4.3.2: *An extract of Table 4.3.8 6-digit combination at time interval 0:20+. The full table can be found in Appendix A.*

OTPs generated between time steps of (0:31+,01:21+)

The data collection continued with intervals of 31, 35 and 40 seconds between two consecutive generated passwords. This can be seen in Table 4.3.11, 4.3.12 and 4.3.13 respectively in Appendix A. At 35 and 40 seconds, there was no sign of repeating combinations or immediate patterns. The intervals were steadily increased to ensure there were no other patterns (Table 4.3.13-4.3.17).

4.3.3 Analysis

1. Pattern

A password repeats itself when one presses the button within < 30 seconds and extending the password time with another 30 seconds. This indicates that the token is TOTP- algorithm.

The column to the right was added to make the tables more organized and to get a clear overview. The column indicates if the password combination was a new one – marked with the number 1 – or if it was a repetition – marked with the number 0. The most repeated pattern at time interval of 20 seconds (Table 4.3.2) is 1 – 1 – 0 – 1 – 1, etc. After one repeated password combination, the token generates one OTP that is unique before generating a password that repeats itself. This shows that decreasing the interval does not result in a new combination, but rather a repeating pattern that is highly predictable most of the time.

Even though the repetition is not constant, meaning the current password is not always the same as the last one, the pattern shows that the repetition factor can still be predicted. The number of repetitions that occurred during generating the password increased with the decreasing time intervals. Other than that, there were no patterns within number combinations or decimal digits.

2. Digit distribution

Of the 516 generated passwords, 478 of them were unique and not a repetition of a previous password. Based on the collected data, it is possible to create an overview of the distribution of the digits used to make the OTP combination produced by DIGIPASS GO3.

Digit	0	1	2	3	4	5	6	7	8	9
Frequency	282	312	313	249	300	279	287	260	293	293

Table 4.3.3: *Table of the frequency of digits used in the password generated by the OTP token provided by DNB. The table shows the distribution of 478 unique password combinations with no repetition.*

	0	1	2	3	4	5	6	7	8	9
a	43	55	49	35	49	44	44	49	55	45
b	60	45	48	46	50	46	49	47	41	46
c	33	68	51	40	45	48	48	45	50	50
d	43	48	51	50	50	49	51	34	57	46
e	49	52	60	41	45	50	48	40	38	55
f	54	44	54	37	51	42	47	45	52	52

Table 4.3.5: *Table of the frequency of digits used in the password generated by the OTP token provided by DNB. The a,b,c,d,e,f represents the position of each digit in a six-digits password, where a represents position one, b position two, and so on.*

Studying the data provided by the tables above, it is not possible to see any noticeable patterns at first glance. The distribution of each digit is more or less the same, with no distinct patterns (Table 4.3.10, Table 4.3.11).

3.Generating OTP

Reconstruction of the algorithm and the way in which the token generates the OTP combination can be further investigated. Since there were no predictable patterns or repeating digits, it is hard to say if the token uses digits as synchronization. If this was an option, it is usually the first digit in every combination, which is used to synchronize the token generated password to the server-generated password.

Based on the published algorithm, it is safe to assume the algorithm used to produce OTPs in this token is based on TOTP. From the experiment, it is known that from 30 seconds and onwards, the token generates new, unique combinations every time regardless of time. From the published algorithm the following formula is used to generate the value of TOTP:

$$Value = HOTP(secretKey, current_T)$$

Set A_0 as an initial value and T_0 as the initial time value. Also assume that T_i is defined as the time the token is pressed to generate another password. When that happens, the generated value can be computed as:

$$Value \ = (X_i)$$

Where X_i represent the six-digit combination.

With the published algorithm in mind, whether the combination is a repetition or not can be determined by the timestep, A_i can be computed as:

$$A_i = \left\lfloor \frac{T_i - T_0}{32} \right\rfloor$$

$$X_i \ = E_k(A_i)$$

The value X_i is encrypted by a function E_k, which depends on a secret key K.

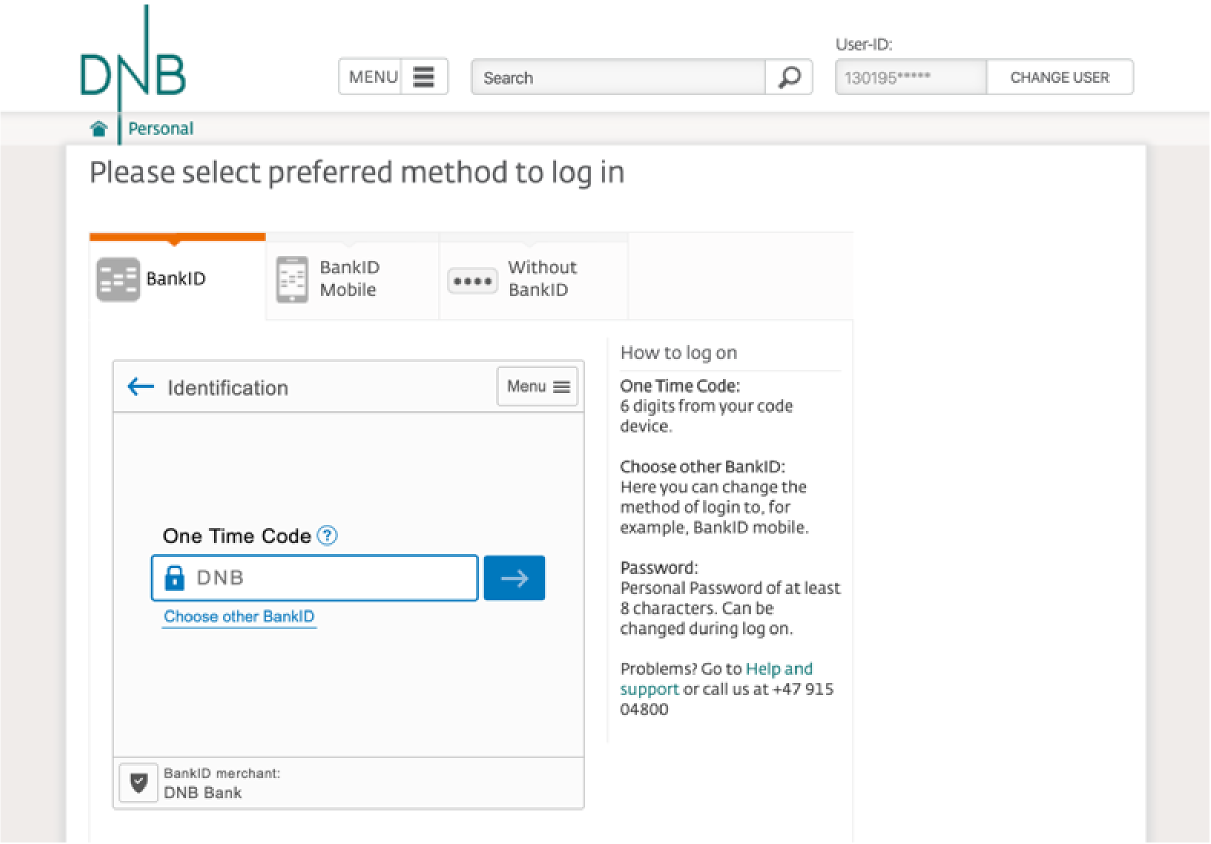
4. Verification

As mentioned earlier, both BankID and without BankID (Uten BankID) uses token-generated OTPs as an authentication method. Since BankID is a common technique for all of the banks in this analysis, the testing of the validation server is based on using BankID as an authentication method to get authorization into the bank system.

To use the generated OTP, one has to use DNB's official website. To select the type of authentication, one has to input an identifier, which is in this case, which in this case is the birth number. The identifier has to be validated before selecting methods or inputting any other personal information. If the identifier does not get accepted by the system, the user receives an error message and this is stuck until inputting a valid identifier that matches DNB's customer database.

Once the identifier gets validated, it is possible to choose between the three types of authentication methods DNB provides (Figure 4). Because the identifier is fixed, and the user can go back and forth through this entire subsite to choose the different ways freely. If a password gets rejected or the authentication process gets rejected by various factors, the user must input the birth number again. The birth number still appears to be fixed on the top right of the user's screen, though the system requires the identifier once more.

Once the method of authentication has been chosen, the user has to enter an OTP. The OTP needs to be validated by the server before proceeding to the next step of inputting a personal passcode. Since the token is based on time, it is assumed that the token has a specific time limit that the generated OTP has to been used within. This means that even though the password is expired on the token, the server can still process the OTP as a valid combination.



User-ID: 130195*****

Please select preferred method to log in

One Time Code: DNB

BankID merchant: DNB Bank

How to log on

One Time Code:
6 digits from your code device.

Choose other BankID:
Here you can change the method of login to, for example, BankID mobile.

Password:
Personal Password of at least 8 characters. Can be changed during log on.

Problems? Go to Help and support or call us at +47 915 04800

Figure 4: *A screenshot of DNB's website once a customer's identifier gets validated. The identifier, which can be seen at the top-right corner, is a fixed factor regardless of which authentication method the user chooses or error messages.*

According to the official HOTP protocol, RFC 4226 « [The authentication protocol] SHOULD NOT be vulnerable to brute force attacks. This implies that a throttling/lockout scheme is RECOMMENDED on the validation server-side». In other words, the protocol recommends systems to have a throttling parameter. This parameter determines how many times a user can get rejected before the account is locked and can no longer be used. DNB has set this parameter to be 3. If the user fails to input either the right OTP combination *or* the personal password three consecutive times, they will be locked out. They cannot use that specific method to get authenticated without contacting the bank to remove the block. However, the user can still choose another approach and get verified, thereby still access their account.

When a user generates an OTP and sends it to the server to authenticate themselves, the server has to validate the OTP input from the user. This happens when the server tries to compute an OTP of its own using a secret key, the same key as the token uses to encrypt the digits and current timestamp. The server produced OTP is then compared to the token generated OTP, and if they are generated within the same step, the OTP gets accepted.

To test this, an OTP of time t was generated and inputted to be authenticated on the server (Table 4.3.6). This attempt was done at different intervals, since the server should handle a delay time from time t, generating an OTP, and t', the time of using it for authentication on the server, so there exist $T > t' - t$.

Time	4:00	3:00	2:00	1:45	1:30	1:29	1:15	1:00
Combination	710604	482303	340426	27712	952734	023307	710892	081080
Status	Rejected	Rejected	Rejected	Rejected	Rejected	Accepted	Accepted	Accepted

Table 4.3.6: *Table of delay handling at the verifier. OTP produced by DIGIPASS GO3 – provided by DNB.*

The table above (4.3.6) states, the T seems to be equal to 1:30 minutes equals to 90 seconds, where all passwords introduced to the system at $t'-t \geq 90$ seconds were rejected. Every OTP produced under 90 seconds got accepted, and the server verified the authentication.

Based on this, the reconstruction of the verification protocol was attempted. Let t be the time of generating an OTP X and t' be the time of using it for authentication on the server. Like the token, the server has to compute an OTP now. This can be formatted as:

$$B = \left\lfloor \frac{t' - T_0}{32} \right\rfloor$$

It has already been proven that $T > t' - t$, where T, in this case, is equivalent to 90 seconds. Since the distance between $t' - t$ is less than 90 second, it means that the token generated OTP was calculated to be in the interval of 10 consecutive numbers as:

$$A = \left\lfloor \frac{t - T_0}{32} \right\rfloor \in \{B - 9, \ldots., B - 1, B\}$$

The verifier computes X'= $E_k(A)$ for every A from the interval. If the encryption of X' is equivalent to X the validation succeeds, and the system accepts the OTP. If not, the OTP is rejected.

5. Possible attack

Let's assume that an attacker gets their hands on a user's identifier and their personal password. Since the pattern of OTPs generated by DNB does not show any specific pattern or irregular use of digits, it is concluded that the probability of guessing one particular OTP is 10^{-6}. This probability depends on if all the ten possible numbers are evenly distributed, which seems to be the case of the password combinations produced by the token provided by DNB.

4.4 Sparebanken Møre

Unlike DNB, Sparebanken Møre is a local bank with a leading market to customers affiliated in Møre og Romsdal [39]. The merging of several local banks in the county resulted in one official brand. Sparebanken Møre, or SBM, is located in every municipality in the county. This is done regardless of the customer demand or the size of the population in each city, based on the key value of being a bank for every individual.

4.4.1 Authentication

Today, SBM offers three different methods of authentication: BankID, BankID on mobile phones, and Password and code (In Norwegian: Passord og kode).

OneSpan

SBM and DNB provide the exact token model, DIGIPASS GO3, to generate OTPs to their customers and is produced by OneSpan. The OTPs generated are produced by a token with serial number 24-0312864-0 (Figure 5).

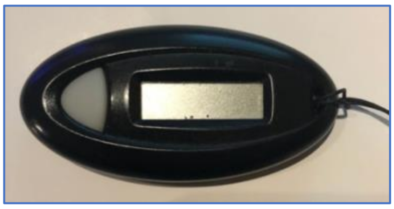


Figure 5: *A picture of token model DIGIPASS GO3 provided by Sparebanken Møre*

To generate an OTP, one has to press the button on the token, where a number consisting of six digits will appear on display. The combination will be displayed on the screen for exactly 40 seconds before disappearing.

4.4.2 Data collecting

1. OTPs generated between time steps of (0:30+,01:05+)

Since the OTP displays on the token for 40 uninterrupted seconds before disappearing, this became a natural starting point for the experiment. 50 six-digit passwords with an interval of 40 seconds were generated, which means the first combination was generated at 0 seconds, the second one at 40 seconds, and the third one after 80 seconds (Table 4.4.7).

Several of the generated combinations are repeating, and a pattern can be observed (Table 4.4.1). For instance, the combination produced at the initial press repeats after 40 seconds. This happens several times with different combinations with an entirely predictable pattern.

1	361309	1
2	361309	0
3	475984	1
4	502253	1
5	502253	0
6	643316	1
7	744955	1
8	744955	0
9	872516	1
10	872516	0
11	942442	1
12	035411	1
13	035411	0
14	156665	1
15	292525	1
16	292525	0
17	381111	1
18	381111	0
19	455117	1
20	516022	1
21	516022	0
22	656521	1
23	656521	0
24	705260	1
25	705260	0
26	857802	1
27	904373	1
28	904373	0
29	036822	1
30	101080	1

Table 4.4.1: *An extract of Table 4.4.7 6-digit combination at time interval 0:40+. The full table can be found in Appendix A.*

The experiment was repeated for different intervals, starting from 30 seconds (Table 4.4.6) and onwards. Like the DNB token, it is possible to regenerate a password before the previous password is outdated by pressing the button twice. The generated sequence at 30 seconds showed that the same password combination repeats itself exactly once before generating a new unique code. There appears to be a significant difference when it comes to the repetition pattern comparing 0:30+ seconds to 0:40+.

By increasing each interval by 5 seconds from 0:40+ to 1:05+ seconds (Tables 4.4.7- 4.4.10, 4.4.13, and 4.4.18), it was easier to determine if the pattern of repetition stays the same, increases or decreases.

As expected, time steps between 0:40+ to 1:05+ showed a difference in the repetition pattern. The repetition goes from occurring after every two new OTPs generated at interval 0:40 to occurring after every three new OTP generated at interval 0:45. The length of these repeated intervals increases in parallel with each increased timestep by five seconds. From time step 0:40 to 0:55 the intervals of each repeated combination increase by one. This means that at time step 0:40 (Table 4.4.7) every 3rd combination is a repetition of the previous password combination. At time step at 0:45 (Table 4.4.8) every 4th combination is a repetition of the last password combination and so on. At timestep 0:55 (Table 4.4.10), the intervals between each repeated digit combination alternate between being seven and eight.

At 01:05 seconds, none of the 40 generated passwords contained repetition. The threshold in which the token can generate a new password every time or repeat a previous digit combination has to be between 01:00 and 1:05. This was easily checked and validated by doing the same experiment between the timesteps 0:59+ and 1:04+ (Tables 4.4.12- 4.4.17). The 0:59+ was included due to the uncertainty of generating enough OTPs at 01:00 to support our initial theory of repetition. At 1:04 no recurrence occurred.

Looking at all the generated data from the timesteps within 0:30 to 1:05, one can easily see that the first digits always are predictable. In every new combination generated, the digit either increases by one or remains the same. The only time the digits remain the same is when the whole combination repeats itself. Since the first digits increase in a gradual manner, it makes each leftmost digit entirely predictable. Therefore, the probability that the first digits increases by 1 modulo 10 after 64 seconds is very close to one. OneSpan has explained that using digits as synchronization is an optional feature in the DIGPASS G03 [40]. This seems to be implemented in this token.

2. OTPs generated between time steps of (01:20:+,05:00+)

The experiment was conducted at larger intervals (Tables 4.4.19 - 4.4.22) to determine if the pattern changes, or if it remains the same. Studying the tables, one will notice certain jumps in the leftmost digits where the first digit. Each combination either increases with a + 1 mod 10, a + 2 mod 10, or a + 3 mod 10 depending on the time interval. The larger the time gap *t* is between generating one OTP from another, the more significant the jump between the first digits of each OTP generated. For example, in Table 4.4.20, the first digit in every combination increases with a + 1 mod 10, a + 2 mod 10. The same pattern continued in Table 4.4.21.

According to the findings, the probability of the first digit increases with a + 2 mod 10 is much larger than a + 1 mod 10. This pattern changes when the time step is increased up to 5 minutes (Table 4.4.22), where the first digits in every combination increase with either a + 4 mod 10 or a + 5 mod 10. Even though it is not predictable to know if the digit increases with either a + 4 mod 10 or a + 5 mod 10, it is safe to say that the probability of the next generated OTP with timestep 05:00 will either be a + 4 mod 10 or a + 5 mod 10 as the first digit is very close to 1.

4.4.3 Analysis

1. Pattern

The accumulated data from the experiments make it possible to construct a general pattern that can be applied to the OTPs generated by DIGIPASS GO3.

1) The smallest time interval used in the measurements and the computations by DIGIPASS and the server is one second.

2) For t time between generating two consecutive OTP where $t \in [64\,\mathrm{a}, 64\,(\mathrm{a}+1))$ and a represents the first digits, the first digit increases by either a or a +1 mod 10 depending on the time step t.

3) If $\mathrm{t} \approx 64 \times \mathrm{a\ seconds}$ the probability p of the first digit increases with a +1 mod 10 is p ≈1 This can be seen in Table 4.4.6 to 4.4.21 where the pattern of first digit increasing by 1 or 0 can be replaced by 1 or 2, where it either increases by a +1 mod 10 or a + 2 mod 10.

2. Digit distribution

In this analysis, 717 passwords were generated in total. 613 of them were new OTP tokens, and 104 were a repetition of a password combination already generated. Looking at the overall frequency of the digits, it is possible to see a clear contrast between the use of numbers when generating a password combination (Table 4.4.2-4.4.3). As seen in the frequency table presented below in Table 4.4.2, the digits 0-5 are used twice as much than the digits 6, 7, 8, and 9.

Digit	0	1	2	3	4	5	6	7	8	9
Frequency	445	455	469	453	419	457	247	232	248	253

Table 4.4.2: *Table of the frequency of digits used in the password generated by the OTP token provided by SBM. The table shows the distribution of 613 unique password combinations with no repetition.*

The assumption that the first digit is used for synchronization was made based on the fact that the first digits are very predictable, and that synchronization is an option at OneSpan. Table 4.4.3 shows an overview of digit distribution used in the last five digits separately and counts how many times they have appeared in each position.

The results, which can be seen in the table below, show that the distribution is not uniform. The same results appeared in the study of OTPs in Sparebanken Vest [2], where a possible reason for this is explained. The last five digits being produced from a 20-bit string of pseudo-random data from the encryption function output. This makes one digit as a 4-bit string, and the rest are represented by a number in the set $\{0,1,2 \ldots,15\}\, mod\ 10$. With the assumption that all 4-bit strings are distributed uniformly, the probability that the digits are 0,1,3,4,5 is 1/8, whereas the probability the digits are 6,7,8,9 is 1/16 [2]. This explanation also seems to be applicable to the collected data represented in the table below.

	0	1	2	3	4	5	6	7	8	9
b	83	76	66	75	85	88	36	30	35	39
c	76	81	84	72	64	84	37	41	31	43
d	68	79	85	91	77	77	28	35	38	35
e	80	79	79	72	63	72	42	27	56	43
f	78	77	96	82	69	78	43	34	28	28

Table 4.2.3: *Table of the frequency of digits used in the password generated by the OTP token provided by Sparebanken Møre. The b,c,d,e,f represents the distribution of digits in the last five positions in a password, where b represents position two, c as position three, and so on.*

3. Generating OTP

Having looked at the collected measurements, it is safe to say that the token is time regulated. Another constant factor is the first digit, which is always increased by 1 modulo 10. Based on the published algorithm and the information known about the general pattern produced by DIGIPASS GO3, the way in which the combination is produced in dissected. Assume A_0 to be an initial value and T_0 as the initial time. T_i is denoted as the time the token is pressed to generate another password. When that happens, the generated value can be computed as:

$$Value \ = \ (a_{i,}X_i)$$

Where $a_{i,}$ represent the first digit and X_i the last five digits.

a_i is calculated by finding T_i the initial time and dividing it with 64, which represents the number of intervals since T_0. This value is reduced to modulo 10 and represents the first digit. The rest of the digits are probably generated based on some secret key, which is most likely the same for both key and server.

$$A_i = \left\lfloor \frac{T_i - T_0}{64} \right\rfloor$$

$$a_i \ \equiv A_i \ mod \ 10$$

$$X_i \ = E_k(A_i)$$

4. Verification

To use a generated OTP, the user has to navigate to the bank's official website and choose BankID to use the password combination. When choosing authentication via BankID, the user has to input their birthnumber as their identifier before entering the OTP and personal password. Since it is known that the token is based on time, it is assumed that the token has to have a specific time limit where the generated OTP has to be used within.

To test this, an OTP of time t is generated (*Table 4.4.5).* By trying different intervals, it is possible to check if the server handles a delayed time from time t, generating an OTP, and t', the time of using it for authentication on the server. It should exist a $T > t' - t$. The testing start by decreasing each interval from 5 minutes. Doing this will help to find the threshold between where an OTP gets accepted by the system and where the connection gets lost.

Time	5:00 +	4:00 +	3:00+	2:00+	1:59+	1:45+	1:30
Combination	640222	390955	715197	254402	991332	711074	457191
Status	Rejected	Rejected	Rejected	Rejected	Accepted	Accepted	Accepted

Table 4.4.5: Table of delay handling at the verifier. OTP produced by DIGIPASS GO3.

As the table above states, the T seems to be equal to two minutes, where all passwords introduced to the system at t’-t ≥ 120 seconds were rejected. Every OTP produced under 120 seconds was accepted, and the server verifies the authentication. A small detail observed during this experiment was that the time of synchronization and confirmation by the server was somewhat longer when the password was introduced at 119 seconds compared to a lower time-interval.

The information above can be used to reconstruct the verification protocol. It has been mentioned that the server computes a separate OTP based on the same shared secret key that the token use and as well as the current timestamp. The server uses this new OTP to compare the user-generated OTP: if they are a match, the validation succeeds.

Let t be the time of generating an OTP a, X, where a represent the first digit and X the remaining five digits of the password combination. Let t’ be the time of using it for authentication on the server. Like the token, the server has to compute an OTP now. This can be formatted as:

$$B = \left\lfloor \frac{t' - T_0}{64} \right\rfloor$$

It has already been proven that $T > t' - t$, where T, in this case, is equivalent to 120 seconds. Since the distance between $t' - t$ is less than 120 seconds, it means that the token generated OTP is calculated to be in the interval of 10 consecutive numbers as:

$$A = \left\lfloor \frac{t - T_0}{64} \right\rfloor \in \{B - 9, \dots., B - 1, B\}$$

Knowing that the verifier computes the first digit a, means the rest of the digits, X', has to be an encryption of A.

$$a \equiv A \, mod \; 10$$
$$X' = E_k(A)$$

If the encryption of X' is equivalent to X the validation succeeds, and the system accepts the OTP. If not, the OTP is rejected.

5. Possible attack

The data analysis highlights that the first number in each combination is highly predictable. Most of the time, the digits increase by a+1 mod 10, and the gap of repetition is consistent with the time intervals. To attack a customer's account, their identifier and personal password needs to be known first. Since the first digit is predictable one can fix a random number as the first digit. For the remaining five numbers: input a random five-digit combination consisting from the set of numbers 0,1,2,3,4,5 in the remaining positions as b, c, d and e. From the digit analysis it is known that the numbers 0 to 5 appear independently with probability 8^{-1}. This means a success probability of guessing a particular password where $X' = X$ is 8^{-5}, which is much higher than the expected probability of 10^{-6}.

4.4.4 Comparing DNB and Sparebanken Møre

There are some differences between **DNB and SBM.** Comparing the data from both DIGIPASS G03 tokens provided by DNB and SBM respectively, it is possible to see that the token has to be modified by a third party. This can be a third-party supplier that is in charge of the production and the functionality of the secret keys internally in the token or the bank itself with a modified implementation, which makes the tokens differ from each other.

Even though the token from DNB had fewer repetitions than the token from SBM, it does not affect the security aspect of OTPs. When the token produces a repeated password combination of an OTP that has already been displayed, it is, by definition, a used password. Therefore, it will not be accepted as a password into the system. While it does not affect the security around the authentication method directly, it can cause problems for a customer to understand the functionality of why a legally produced password is not working.

Another highly noticeable difference between the two tokens is the first digit of each produced OTP combination. While there is no sign of a noticeable pattern at first glance when analyzing the token provided by DNB, the first digit of all password combinations from SBM were highly predictable. This also highlights that the bank's internal security systems have a significant part to play in adapting and modifying the tokens into their security protocols and methods.

4.5 Sparebank 1 Nordvest

Sparebank 1 Nordvest is a bank that is part of an alliance between 14 independent banks that works under the same platform and brand, Sparebank 1 [41]. The bank, which has its headquarters in Oslo, is the second-largest bank in Norway with over 352 branches. Sparebank 1 Nordvest acts as a local bank within the county and offers a wide range of services within financial services, both for retail and corporate customers [42].

4.5.1 Authentication

Sparebank1 Nordvest offers three types of methods for authentication: BankID, BankID on mobile phones, and Without BankID. As it was mentioned earlier with DNB: Without BankID is not a method provided ID-porten, but an authentication method operated by the bank. It is thus not possible to use a token provided by another bank to apply for user authentication. In addition to an identifier, the user has to input an OTP and a personal password to be authenticated and get access to their account.

Gemalto

Sparebank 1 uses a lightweight token produced by Gemalto. All the data in this analysis is collected using two tokens in the model Gemalto Lava token with the serial numbers 15749002010951131308 and 22919068000018921712. All the independent banks in the Sparebank 1 alliance provide the same token model to their customers.

Figure 6: *A picture of token model LAVA Token provided by Sparebank 1 Nordvest*

To generate an OTP, one has to press the button on the token, and a number consisting of six digits will appear on the LCD-screen. The combination will be displayed on the screen for exactly 29 seconds before disappearing. When one tries to regenerate a password 29 seconds after the initial press, a new OTP will appear on display. Unlike the other tokens that are mentioned in this analysis, it is not possible to press the button twice to regenerate an OTP combination. The combination has to disappear from the screen before one can generate a new one. This is a feature that is not mentioned in Gemalto, which is odd since they have descriptive explanations of every feature each token provides.

4.5.2 Data collecting

1. OTPs generated between time steps of (0:29+,0:31+)

Since the shortest possible interval between each generated password is 29 seconds, the experiment starts by generating 50 OTP combinations with an interval of 29 seconds (*Table 4.5.1)*. It is observed that every password starts with digit number 9. There are some minor repetitions of the digit in the second position in the combination, but this happens irregularly. The same experiment is repeated with intervals of 30 (*Table 4.5.7)* and 31 (*Table 4.5.8)* seconds between each press to check any possible changes in the results.

1	949231	1
2	943887	0
3	916614	1
4	956587	1
5	939699	1
6	938677	0
7	988474	1
8	922805	1
9	999131	1
10	936142	1
11	921632	1
12	985529	1
13	939087	1
14	906001	1
15	901201	0
16	975156	1
17	938204	1
18	989291	1
19	982973	0
20	940510	1
21	931607	1
22	909416	1
23	983344	1
24	918208	1
25	919829	0
26	920328	1
27	941728	1
28	931987	1
29	972092	1
30	976798	0

Table 4.5.1: *An extract of Table 4.5.6 6-digit combination at time interval 0:29+ The full table can be found in Appendix A.*

The table above shows that the first digit is repeated in all of the combinations. Second digits repeat in some places. This is possible to see in $i = 5$ and $i = 6$ with combination 939699 and 938677 and in $i = 14$ and $i = 15$ with combinations 906001 and 901201. Since the first digit is repeated, the rightmost column is marked as 1 when there is no repetition and 0 when the second decimal digit in the OTP combination repeats itself.

2. OTPs generated between time steps of (0:35+,05:00+)

When increasing the interval from 35 seconds, 40 seconds to 1 minute, the collected data remained unchanged. The token produced a new combination every time it was pressed. The combinations can be seen under *A3 Sparebanken 1* (Table 4.5.9-4.5.15). The earlier experiments conducted with the tokens provided by DNB and Sparebanken Møre have shown that severe changes happen between the threshold of 29 and 30 seconds. Based on this, further testing was done at intervals of 01:29 and 1:30, as well as 2 minutes and 5 minutes to see if a more significant gap in intervals had any impact on the OTPs (Table 4.5.11- 4.5.15).

4.5.3 Analysis

1. Pattern

Every single password combination generated by Gemalto Lava token start with digit number 9. There appears to be more repetition of the second digits within the first 20 intervals than the rest. This pattern becomes more evident the more prolonged the intervals between each new OTP produced. Looking closer at the timesteps at 60 seconds (Table 4.5.11), 89 seconds (Table 4.5.12), and 90 seconds (Table 4.5.2 and 4.5.13), it is clear that the repetition pattern occurs more in the first 20 intervals than the rest. Since the repetition does not increase or decrease parallel with each timestep, it is hard to say if this is a mere coincidence or if it has an underlying meaning without a more definite pattern.

1	947986	1	11	983197	1	21	916522	1	31	920155	1
2	921023	1	12	970971	1	22	982320	1	32	945998	1
3	957696	1	13	938338	1	23	901754	1	33	968654	1
4	955120	0	14	963685	1	24	953071	1	34	991259	1
5	957678	0	15	994757	1	25	900106	1	35	915580	1
6	962020	1	16	998215	0	26	915766	1	36	920598	1
7	976724	1	17	922349	1	27	916157	0	37	903432	1
8	969650	1	18	946803	1	28	967332	1	38	934470	1
9	965730	0	19	989686	1	29	944608	1	39	974650	1
10	945699	1	20	987288	0	30	939714	1	40	933501	1

Table 4.5.2: *An extract of Table 4.5.13 6-digit combination at time interval 1:30+. The table shows the difference in repetition pattern regards to the second decimal digits. One can see a difference in repetition pattern for the first 20 password combinations compared to the last 20 password combinations. The full table can be found in Appendix A.*

2. Digit distribution

Altogether, 433 unique password combinations were generated by the Gemalto Lava token provided by Sparebank 1 Nordvest. Looking at the digit distribution, one can see there is an even distribution of numbers except for digit number 9 (Table 4.5.3 - 4.5.4). This is expected since all the password combinations starts with 9. This number also appears in other positions in the 6-digit OTP combination.

Digit	0	1	2	3	4	5	6	7	8	9
Frequency	220	223	212	211	217	203	230	229	221	632

Table 4.5.3: *Table of the frequency of digits used in the password generated by the OTP token provided by Sparebank 1 Nordvest. The table shows the distribution of 433 unique password combinations with no repetition.*

	0	1	2	3	4	5	6	7	8	9
a	0	0	0	0	0	0	0	0	0	433
b	46	47	39	54	54	36	44	48	38	27
c	43	40	33	53	34	44	41	42	53	50
d	37	46	40	34	44	39	53	53	49	38
e	45	40	61	36	39	47	39	45	38	43
f	49	50	39	34	46	37	53	41	43	41

Table 4.5.4: *Table of the frequency of digits used in the passwords generated by the OTP token provided by Sparebank 1 Nordvest. The a, b, c, d, e, f represents the position of each digit in a 6-digits password, where a represents position one, b position two, and so on.*

3. Generating OTP

A constant factor that appears in all the password combinations is the leftmost digit in every password. Out of pure curiosity, the following questions were sent to some acquaintances asking them these three specific questions.

1) Are you a customer in any of the banks that are marketed under Sparebank 1 Alliansen?
2) If their bank distributed Gemalto Lava token as a device for generating OTPs?
3) If every password combination started with number 9?

Based on the four people surveyed it was assumed that every password combination produced by the Gemalto Lava token provided by Sparebank 1 Alliansen most likely use digit number 9 as the leftmost decimal digit. The acquaintances who responded were customers of Sparebank 1 Østlandet, Sparebank 1 SMN, and Sparebank 1 SR-Bank.

The way the combination is produced was then reconstructed. Let A_0 be an initial value and T_0 the value of the initial time. Also let T_i represent the time the token is pressed to generate another password. When that happens, the generated value can be computed as:

$$Value \ = (9, X_i)$$

$$A_i = \left\lfloor \frac{T_i - T_0}{Tx} \right\rfloor$$

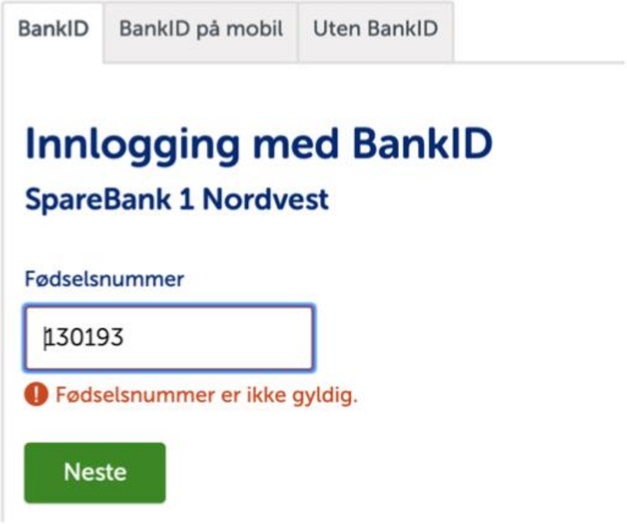
$$X_i \ = E_k(A_i)$$

Where 9 represents the first digit, X_i the last five digits and Tx time step between the generated OTPs produced by the same token. This is believed to be 29 seconds since the token generates a new password every 29 seconds. The rest of the digits are probably encrypted based on some secret key, which is most likely the same for both the key and server.

4. Verification

Navigating through Sparebank 1 official website, the user has to select one of the 14 banks they are a customer of. It is possible to choose a local bank other than their own. Still, the identifier will not get authenticated as every local bank has their own database of customers - even though they are established under the same platform. When choosing authentication via BankID, the user has to input their bith number as their identifier. The identifier has to be validated, before entering the OTP and their personal password (Figure 7). Whereas the OTP is accepted and verified by the server becomes a known factor for the user after inputting the private password. When rejected, the server redirects to the homepage, where the user gets a chance to do the process all over again.

One noticeable factor is that when an identifier gets validated by the system, but the user uses some time to input the OTP, the page will automatically redirect to the homepage. This happens exactly after 5 minutes. When this happens, one gets redirected to the website of Sparebank 1 Nordvest again and has to choose between authentication methods. The system also requires the user to input their identifier for validation again. Nevertheless, the experiment still shows that one can get the identifier verified for the second time using the same OTP. The authentication process succeeds, given it is within the allowed time limit.



Innlogging med BankID

SpareBank 1 Nordvest

Fødselsnummer

130193

Fødselsnummer er ikke gyldig.

Figure 7: *A screenshot of the pop-up window at Sparebank 1 Nordvest, where the user's inputted identifier does not get validated. Note that the birthnumber shown in the picture is not completed and has been censored for privacy reasons.*

The initial starting point was at 5 minutes. This interval got accepted, so the time value was increased to 12 minutes. Once it got rejected, the timestep was decreased by two minutes each time to find the threshold of where an OTP got accepted by the system and where the connection got lost.

Time	12:00+	10:00+	8:00+	7:59 +	7:45 +	7:30+	7:29	7:15	7:00+	6:00+	5:00
Combination	907834	990217	971650	964412	960396	939620	976173	939620	974270	961793	986641
Status	Rejected	Rejected	Rejected	Rejected	Rejected	Rejected	Accepted	Accepted	Accepted	Accepted	Accepted

Table 4.5.5: Table of delay handling at the verifier. OTP produced by Gemalto Lava token – provided by Sparebank 1 Nordvest.

As Table 4.5.5 states, the T seems to be equal to 8 minutes or 480 seconds, where all passwords introduced to the system at t'-t ≥ 480 seconds got rejected. Every OTP produced with 479 seconds or less got accepted, and the server verified the authentication.

The server verification protocol can be reconstructed as following : Set t be the time of generating an OTP $9, X$, where 9 represents the first digit and X the remaining five digits of the password combination.

Let t' be the time of using it for authentication on the server. Like the token, the server has to compute an OTP now. This protocol is similar to the one constructed and presented in the previous section about the server verification in 4.4 Sparebanken Møre, where the only difference is the timestep and the first digit. 9 is always a constant number used for synchronization, which means the rest of the digits are computed:

$$X' = E_k(A)$$

If the encryption of X' is equivalent to X, the validation succeeds, and the system accepts the OTP. If not, the OTP is rejected.

5. Possible attack

From the collected data it is seen that the first number in each combination is constant and does not change regardless of timesteps. To attack a customer account, one first needs to know their identifier and personal password. Since the first decimal digit is fixed, the attacker has only five more digits to brute-force. This also means that the probability of guessing the remaining five passwords is 10^{-5}, which is higher than the expected probability of 10^{-6}

4.6 Nordea

Nordea is a result of merging many local banks within Finland, Denmark, Sweden and Norway and was officially founded in 2000 [43]. With its headquarters in Helsinki, Finland, the bank has more than 10 million users worldwide and ranks as one of the top 10 banks in Europe. Nordea offers a wide range of financial services, within everything from banking and insurance. The bank is also one of the few banks in Europe that gives AA-rating, meaning the bank offers second-highest bond rating by Standard & Poor's Credit Market Services Europe Ltd (''**S&P**'') [44].

4.6.1 Authentication

Nordea offers three methods of authentication: BankID on mobile phones, BankID, and Basistjenesten (roughly translated as Basic Service). Basistjenesten is quite similar to the authentication method MinID provided by ID-porten but does not require an OTP to get authenticated. The method is solely based on an identifier and a personal password, which makes it less secure than the other techniques that Nordea offers, or the other techniques covered in this study. Nordea also clarifies that this is an authentication method specially developed for children under the age of 15 who require access to internet banking. The minimum age requirement of accessing BankID and BankID on mobile phones is 15 [45]. The method also gives limited authorization to the account and revoke certain privileges if the user is underaged.

HID Global

Nordea provides a HID- pocket token to their customers to generate passwords for authentication. Look-wise, this pocket token is more complicated than the rest of the tokens presented and analyzed in this study. To use this token, one has to input a four-digit PIN. This is done by pressing the green power button (See Figure 8) on the bottom right. To generate the password, one needs to press a four-digit PIN-code using the keypad embedded in the token. This personal code provides an added security layer and a personalized possession factor.

The primary testing is done on a token with serial number 0967681345, issued MAY-17 (Pictured under in figure 8). In the analysis of server delay and verification this token is compared to another token with the serial number 0973676606, issued OCT-17.

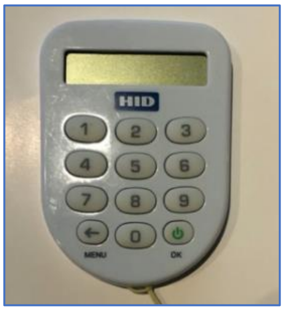


Figure 8: *A picture of token model HID Pocket token provided by Nordea*

Once the PIN-code is entered, an 8-digit OTP combination will appear on the screen. The combination will be on display for precisely 55 seconds before it disappears. When one tries to regenerate a password again, the PIN-code will be required. It is possible to press the power button within the 55 seconds and input the PIN again to renew another password.

4.6.2 Data collecting

1. OTPs generated between time steps of (0:15+,0:55+)

Since the OTP stays for 55 seconds on the LCD-screen, the natural step was to generate a password with a timestep of 55 seconds (Table 4.6.1). It appears to be a constant pattern of repetition for the first decimal digit in the password combination. The second digit is also predictable. The time step is decreased to intervals with 30 seconds to check if the pattern changed or not.

The first digit increases steadily by adding + 1 mod 10 to the previous digits or otherwise remains the same. The second digit also increases by one in every digit combination. T

To examine the first digit further, the time steps are shortened to 15, 25, 40, and 55 seconds (*Table 4.6.6-4.6.9*). Theoretically, it is possible to reduce the interval further down to 10 or even 5 seconds, but to generate, write down the generated combination and generate a new OTP combination by turning off and on the token and then input the personal PIN code would take about 12 seconds. To avoid any fluctuation and keep the experiment as precise as possible, the minimum time interval presented in this thesis is 15 seconds.

The table below shows (Table 4.6.1) the password combination collected at time step of 55 seconds. The rightmost column is marked with 0 if the first digit is the same as the previous combination, or as 1 if it is increased by one.

It can be observed that the shorter the interval gets between two generated OTP combination, the larger the number of repetitions of the first digit occurs. The interval length is increased to check if the repetition occurs less, which it does steadily.

1	78642232	1
2	79225849	0
3	80342380	1
4	91979605	1
5	92224725	0
6	93512032	0
7	04354136	1
8	05405406	0
9	16558333	1
10	17346803	0
11	28203103	1
12	29567917	0
13	20730454	0
14	31544032	1
15	32141521	0
16	43056824	1
17	44239362	0
18	55481303	1
19	56572150	0
20	67408103	1
21	68085832	0
22	69098430	0
23	70331130	1
24	71913503	0
25	82134280	1
26	83438296	0
27	94300602	1
28	95200008	0
29	06297201	1
30	07004463	0

Table 4.6.1: *An extract of Table 4.6.8 9-digit combination at time interval 0:55+ The full table can be found in Appendix A.*

If one generates an OTP combination and gets a second digit d, it is possible to determine the second digit of the next OTP to be $d + 1\ mod\ 10$. The time in between generating these combinations can vary from five seconds to five days, but the continuance is constant. For example, the above table (Table 4.6.1), shows that the first generated OTP, where $i = 1$, is equal to 78642232. The next password generated is $i = 2 =$ 79225849. This pattern continues throughout the entire table and proceeds to the next table and makes the second digit in every password combination produced by the token very predictable. Since the collected data is presented in descending order in Appendix A, it is not possible to see the continuation of the second digit in its pure form. In addition to the second digit, the first digit is also always predictable.

2. OTPs generated between time steps of (0:59+,2:00+)

When trying intervals between 59 seconds to 2 minutes (*Table 4.6.10-4.6.16*), it is possible to see that the gaps between the repetition of the first digit gets larger parallelly with the intervals. In every new combination generated the digit either increases by one or stays the same. Since the first digit increases gradually, it makes each leftmost digit entirely predictable. Therefore, the probability of the first digits increases by one modulo 10 after two minutes very close to one. The gap between each repeating sequence widens parallelly with the intervals.

3. OTPs generated between timesteps of (5:00+,10:00+)

From Tables 4.6.16 to 4.6.19, it is possible to notice certain jumps in the leftmost digits. The first digit of each combination either increases with $a + c \, mod \, 10$ or where c equals two or more can be observed. As the additional variable c gets larger, the more significant intervals get between two consecutive generated OTP combinations. At timestep 5 and 6 minutes (Table 4.6.17 and Table 4.6.18) and onwards the first digit increases with $a + 2 \, mod \, 10$ or $a + 3 \, mod \, 10$, and at 8 and 10 minutes (Table 4.6.19-4.6.20) $a + 4 \, mod \, 10$.

4.6.3 Analysis

1. Pattern

From the accumulated data it possible to construct a general pattern that can be applied to the OTPs generated by the HID Pocket token.

1) For t time between generating two consecutive OTP, where t is unlimited, the second digit d increases with $d + 1$.
2) For t time between generating two consecutive OTP where $t \in [120 \text{ a}, 120 \text{ (a } + 1))$, and a represents the first digit, the first digit increases by either a or $a + 1 \, mod \, 10$ depending on time step t.
3) If $\text{t} \approx 120 \times \text{a seconds}$ the probability p of the first digit increases with $a + 1 \, mod \, 10$ is p ≈1. This can be in Table 4.6.6 to 4.6.20, where the pattern of the first digit is increasing by 1 or 0 can be replaced by 2, 3 and 4, where it either increases by $a + 1 \, mod \, 10, a + 2 \, mod \, 10$, $a + 3 \, mod \, 10$ and $a + 4 \, mod \, 10$.

2. Digit distribution

550 unique password combinations were collected and generated by the HID Pocket token provided by Nordea. Looking at the overall frequency of the digits, it is possible to see a clear contrast between the use of numbers when generating a password combination (Table 4.6.2-4.6.3). In the frequency table presented below in Table 4.6.2, the digits 0-5 are used almost twice more than 6, 7, 8, and 9.

Digit	0	1	2	3	4	5	6	7	8	9
Frequency	513	523	511	507	528	549	318	334	311	306

Table 4.6.2: *Table of the frequency of digits used in the password generated by the OTP token provided by Nordea. The table shows the distribution of 550 unique password combinations with no repetition.*

	0	1	2	3	4	5	6	7	8	9
a	54	48	54	47	63	54	58	63	55	54
b	53	55	56	56	54	56	55	54	56	55
c	74	61	78	56	70	69	32	38	27	45
d	73	75	60	72	61	77	36	34	31	31
e	56	77	73	74	62	80	37	38	32	21
f	57	66	67	62	71	81	30	44	42	30
g	78	69	70	73	65	66	30	31	31	37
h	68	72	53	67	82	66	40	32	37	33

Table 4.6.3: *Table of the frequency of digits used in the password generated by the OTP token provided by Nordea. The a,b,c,d,e,f,g,h represents the position of each digit in an 8-digits password, where a represents position one, b position two, and so on.*

The results in Table 4.6.3 shows that the distribution is not uniform. The two first positions are distributed evenly, which is consistent with the patterns found during data collecting. The first and second digit is following a preprogrammed pattern, meaning there are no differences between each number when it comes to distribution, even though the pattern is predictable. From position c and outwards, there is a clear distinction between the use of the digits 0-5 and 6, 7, 8, and 9. The probability of a combination using the digits 0-5 close to 1/8, which is much higher compared to the rest of the numbers with probability of 1/16.

3. Generating OTP

HID pocket tokens appears to have two factors that are highly predictable. The first digit is increasing steadily and the second digit is $a + 1 \ mod \ 10$ of the previously generated OTPs second digits. The tokens also seem to have an inbuilt counter (figure 9) and watch, which makes it hard to decide if the token is event or time-based. The counter value increases with each OTP generated by the token.

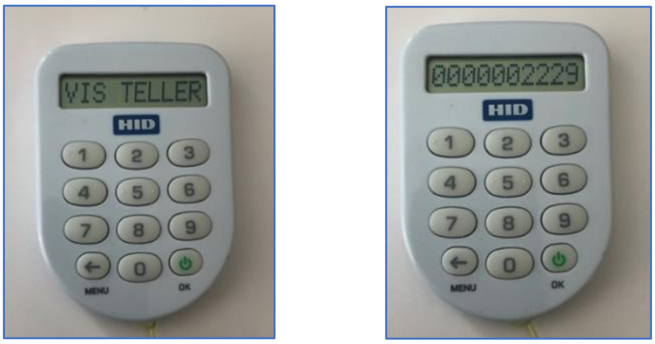


Figure 8: *A picture of the token model HID Pocket token provided by Nordea and the inbuilt counter function. It is possible to see that this particular token has in total generated 2229 passwords.*

Based on the published algorithm and the information collected during experimenting, the way a combination is produced is reconstructed. Let A_0 be an initial value and T_0 the value of the initial time. Also denote T_i as the time the token is pressed to generate another password. When that happens, the generated value can be computed as:

$$Value = (a_{i,}\, d_{i,} X_i)$$

Where $a_{i,}$ represent the first digit, d_i the second digit and X_i the last six digits of the combination.

a_i is calculated by finding the time between the initial time and password generated at time T_i and dividing it with 120 seconds, which represents the number of intervals since T_0. This value is reduced to modulo 10 and represents the first digit. The second digit is found by increasing the value of the second decimal digit of the previous password by $d\ + 1\ mod\ 10$.

The rest of the digits are probably encrypted based on function E_k for some secret key K. This is most likely the same for both the key and server.

$$A_i = \left\lfloor \frac{T_i - T_0}{120} \right\rfloor$$

$$a_i \equiv A_i\ mod\ 10$$

$$d_i \equiv d_{i-1} + 1\ mod\ 10$$

$$X_i = E_k(A_i, d_i\ )$$

4. Verification

The only token-based authentication method through Nordea is by using BankID. To do this, the user needs to navigate to Nordea’s website, choose to login and then choose a method of authentication. When choosing authentication via BankID, the user has to input their birth number as their identifier, before entering the OTP and personal password. This must get validated. Thus, to gain access to the system, one needs to have a valid identifier, regardless of whether they are customer of Nordea. The user inputs both OTP and personal password separately, and each has to get validated individually before getting authenticated.

Since it was not known if the token was based on time and event-based algorithm, it was checked how the server handles the delay. This was done by generating an OTP of time t and inputting this for authentication on the server (*Table 4.6.4)*. By trying different intervals, it is possible to check if the server handles a delayed time from time t, generating an OTP, and t’, the time of using it for authentication on the server. There should exist a $T > t' - t$. The intervals are increased from 5 minutes. Doing this will help finding the threshold between the time where an OTP gets accepted by the system and where the connection gets lost.

Time	5:00+	10:00+	30:00 +	60:00 +	120:00+	360:00+	420:00+
Combination	39446441	60032883	41322120	3204408	43651220	00002226	79020430
Status	Accepted	Accepted	Accepted	Accepted	Accepted	Accepted	Accepted
	480:00+	720:00+	839:00+	840:00+	900:00+	960:00+	1440:00+
	49106352	86442463	14255886	81137329	14255886	49402736	80375115
	Accepted	Accepted	Accepted	Rejected	Rejected	Rejected	Rejected

Table 4.6.4: Table of delay handling at the verifier. OTP produced by HID Pocket token – provided by Nordea.

As the table above states, the T seems to be equal to 14 hours, where all passwords introduced to the system at $t' - t.$ ≥ 840 minutes were rejected. Every OTP produced under 14 hours got accepted, and the server verified the authentication. One noticeable factor is that the loading time got longer the more prominent the parameter T was.

Based on this information reconstruction was attempted. This can be formatted as:

$$B = \left\lfloor \frac{t' - T_0}{120} \right\rfloor$$

Since the distance between $t' - t$ is less than 840 minutes, it means that the token generated OTP is calculated to be in the interval of 10 consecutive numbers as:

$$A = \left\lfloor \frac{t - T_0}{120} \right\rfloor \in \{B - 9, \ldots., B - 1, B\}$$

Assuming the second digits works as a counter and the verifier computes the first digit a, means the rest of the digits, X', has to be an encryption of A.

$$a \equiv A \, mod \, 10$$
$$X' = E_k(A, d)$$

If the encryption of X' is equivalent to X the validation succeeds, and the system accepts the OTP. If not, the OTP is rejected.

Since T is much larger compared to other tokens presented in this thesis, the result was discussed with an acquaintance who was a customer of the same bank. The discussion led to another round of experiments based on the server delay.

Time	5:00+	10:00+	20:00 +	29:59	30:00 +	60:00+
Combination	75780081	26193182	38491452	59939246	77222044	13036141
Status	Accepted	Accepted	Accepted	Accepted	Rejected	Rejected

Table 4.6.5: Table of delay handling at the verifier. OTP produced by HID Pocket token – provided by Nordea.

The table shows that the T is much smaller than the previous token. The predominant factor that differentiates these two tokens from each other is when they are produced. The first tested token was produced in May-17, whereas the last one in October-17.

5. Possible attack

The first and second digit are highly predictable. When the OTP is produced with intervals of two minutes, the probability of the first digits increase by a+1 mod 10 is close to 1. The second digit works as a counter and is a constant moving factor that increases steadily. Assume that the customer's identifier, personal password and the four-digit token pin is known to the attacker. Assume remaining six digits are picked any random digit combination from the set of numbers 0,1,2,3,4,5 in the remaining positions as c, d, e, f, g and h. From the digit analysis it is known that the numbers 0 to 5 appear independently with probability 8^{-1}. This means a success probability of guessing a particular password where $X' = X$ is 8^{-6}, which is much higher than the expected probability of 10^{-8}.

Chapter 5

Discussion and Conclusion

5.1 Results

The experiment was conducted to see if tokens generating one-time passwords matched with the expected security standards. Through examining and experimenting a large amount of data generated from four different tokens, predictable patterns and repeating combinations were discovered. From each analyzed token, it is possible to see a number of weaknesses that could have damaging effects. This study has also led to more questions about the protocols, how the algorithm is fixed, and if the algorithm is decided internally by each bank, a third-party supplier, or by the manufactures that produced the tokens.

5.1.1 Comparing Results

DIGIPASS GO3

From the analyzation, it is possible to see some similarities between the tokens analyzed in this thesis. First of all, the patterns between Sparebanken Vest and Sparebanken Møre are quite similar. Comparing Semaev's research [2] on possible patterns and internal algorithm reconstruction, it is relatively easy to see that it is almost identical to the token analyzed from Sparebanken Møre. Semaev [2] has also concluded that the success probability of an attack is 8^{-5}, which is equal to the probability found in the analysis of Sparebanken Møre.

The possible reason for the results being almost identical can be because Sparebanken Vest and Sparebanken Møre are both local banks affiliated within a specific region. Similar to Sparebanken Møre, Sparebanken Vest is a result of the merging of several local banks within a particular geographical area [46]. Even though both of their websites have different design approaches, both offer the same services to their customers regarding financial services, methods of authentication and token models. These elements indicate that the bank's business model could be based on the same principles, which leads back to the theory of a third-party being involved.

This theory was discussed back in chapter 4.4.4 Comparing DNB and Sparebanken Møre, as both of the banks offer the token model DIGIPASS GO3 to their customers. DNB had no predictable patterns and a uniform digit distribution, which is a massive contrast to the results found with the token provided by Sparebanken Møre. From the security and functionality aspects, there are no similarities between the two independent experiments other than having a token manufactured by VASCO/OneSpan. However, if a third party is in-charge of the production and the functionality of the secret keys internally in the token, the supplier could have the opportunity to apply the same algorithm to all of their customers.

Possible attack

A dynamic password is expected to be unpredictable, where the digits are chosen randomly. Therefore, the optimal probability is expected to be 10^{-6} or 10^{-8} depending on the length of the password combination and if all ten possible numbers are evenly distributed. According to the findings in this study, this is clearly not the case. In total, three out of four tokens in this study indicate weaknesses, which increases the expected success probability of an attack to 10^{-5}, 8^{-5} and 8^{-6}. The token provided by DNB is the only token that does not show any sign of predictable patterns or uneven digit distribution.

The tokens from Sparebanken Møre and Sparebank 1 reveals that the first digit in every combination is highly foreseeable. However, differences regarding the digit distribution makes the success probability different from each other. The frequency table based on token provided by Sparebanken Møre (Chapter 4.4.3 Analysis) shows that the digits distribution is not uniform. The numbers 0,1,2,3,4,5 are almost used twice as much as the remaining digits 6,7,8,9, which leads to a success probability of an attack to 8^{-5}.

On the contrary, the collected data from Gemalto Lava token provided by Sparebank 1 does not show any sign of uneven digit distribution. Even though every password combination starts with digit number 9, this does not affect the digit distribution of the remaining five digits. This also means that the probability of an attack success is 10^{-5}, which is higher than the expected probability of 10^{-6}.

Moreover, based on Nordea and the HID-pocket token they provide, it is possible to see that the first two digits of the eight-digit OTP combination are predictable. The collected data (Table 4.6.5-4.6.19 in Appendix A) shows that the first digit increases steadily according to the time interval between two consecutive OTPs. The second digit works as a counter. As a result, the digit in the second position increases by one for each generated password. Since the remaining six digits are not distributed uniformly, this increases the expected success probability from 10^{-8} to 8^{-6}.

5.1.2 Improvements

Fluctuations in data

Since the data collecting was a one-man job, there could be fluctuations in the data. The given value of time in the tables is approximate to the real-time of pressing and generating the OTP. This issue is also referred to in 4.2.2 Data Colleting. Some of the time intervals were experimented at several times because of non-consistent patterns. Even though fluctuations are minimal, there could be some inconsistencies within small parts of the tables and the constructed algorithm that should apply as a standard formula for the collected data. This could have been avoided with proper manpower or more effective equipment.

Different attacks

Time is an unpredictable element. It is not possible to forecast how much time an experiment could take. This was also experienced while conducting the experiments and writing this thesis. Many rounds of data collection had to be performed in order to maintain the quality and stability of the data presented in this study, which took more time than expected. In Semaev's experimental study of DIGIPASS GO3 [2], there were attacks presented where one or many customers were targeted. By including a similar analysis of possible attacks could have given this thesis more depth. Overall, integrity and quality of data was more important than analyzing numerous attacks on different levels.

5.2 Recommendations for Further Work

Experiments based on token-based OTPs

As this experimental study only focused on one authentication method, it would be interesting to investigate further on other methods that requires token generated one-time passwords. It is also known from Chapter 3.1 that one-time passwords can be produced by downloadable mobile apps developed by the respective banks. Comparing the security aspects of password combinations generated from hard-tokens and applications could be continued by other researchers.

BankID on mobile phones

A popular alternative for normal BankID is BankID on mobile phones. As presented in Chapter 4.2.2, this method only requires an identifier and a cellphone with an activated SIM for BankID on mobile. Experimenting with this method of authentication by studying signals sent to and from the phone would be recommended. An analysis of signals could lead to finding predictable patterns and possible attacks.

This would also create a better understanding of the underlying algorithms and potential weaknesses in the system.

5.3 Conclusion

The main aim of this study was to understand how a token generated password work and determine if the expected security standards for an attack are upheld. By collecting a large number of password combinations produced by four different hard tokens at different intervals, it was possible to construct each internal token algorithm to some extent, in addition to analyzing the server delays.

This research highlights a number of weaknesses in the different tokens provided by DNB, Sparebanken Møre, Sparebank 1 and Nordea. It is possible to notice that the expected probability of 10^{-6} or 10^{-8} is not upheld. The obtained results show predictable patterns that are inconsistent with the expected requirements of a dynamic one-time password. This is also theoretically proven by explaining a basic attack.

In conclusion, this study points out that the probability of predicting a password is much higher than the expected rate. A dynamic one-time password is meant to strengthen the authentication, but the weaknesses found in this study could cause problems if one or several customers are targeted over an extend period of time.

Appendix A

A1 DNB

Table 4.3.7: *6-digit combinations at 0:15+ generated by DIGIPASS 3 – provided by DNB.*

1	766537	1
2	766537	0
3	462535	1
4	462535	0
5	181894	1
6	181894	0
7	471303	1
8	471303	0
9	177941	1
10	177941	0
11	791547	1
12	791547	0
13	791547	0
14	461596	1
15	461596	0
16	148443	1
17	148443	0
18	141515	1
19	141515	0
20	593654	1
21	593654	0
22	745324	1
23	745324	0
24	997093	1
25	997093	0
26	136927	1
27	136927	0
28	926970	1
29	926970	0
30	926970	0

Table 4.3.8: *6-digit combinations at 0:20+ generated by DIGIPASS 3 – provided by DNB.*

1	336448	1
2	329342	1
3	329342	0
4	835663	1
5	997727	1
6	997727	0
7	529944	1
8	338560	1
9	338560	0
10	868048	1
11	868048	1
12	269993	0
13	824128	1
14	824128	0
15	204037	1
16	908755	1
17	908755	0
18	611599	1
19	611599	0
20	008277	1
21	485641	1
22	485641	0
23	651281	1
24	324823	1
25	324823	0
26	841999	1
27	448676	1
28	448676	0
29	964969	1
30	964969	0
31	291379	1
32	133592	1
33	133592	0
34	571055	1
35	571055	0
36	640924	1
37	101857	1
38	101857	0
39	990575	1
40	121108	1
41	121108	0
42	145989	1
43	145989	0
44	159112	1
45	295541	1
46	295541	0
47	481446	1
48	783567	1
49	783567	0
50	340390	1

Table 4.3.09: *6-digit combinations at 0:29+ generated by DIGIPASS 3 – provided by DNB.*

1	263593	1
2	902074	1
3	095026	1
4	722884	1
5	465915	1
6	716970	1
7	716970	0
8	969617	1
9	198764	1
10	151325	1
11	937176	1
12	142807	1
13	648512	1
14	094850	1
15	920328	1
16	581245	1
17	033822	1
18	033822	0
19	477610	1
20	650619	1
21	899844	1
22	276329	1
23	119578	1
24	043407	1
25	448982	1
26	246106	1
27	671218	1
28	487260	1
29	487260	0
30	581633	1
31	012799	1
32	672199	1
33	494002	1
34	505106	1
35	154582	1
36	508873	1
37	372142	1
38	438402	1
39	564619	1
40	564619	0
41	472918	1
42	073913	1
43	757817	1
44	657465	1
45	826595	1
46	135912	1
47	505226	1
48	564619	1
49	826106	1
50	681788	1

Table 4.3.10: *6-digit combinations at 0:30+ generated by DIGIPASS 3 – provided by DNB.*

1	427467	1
2	434511	1
3	752615	1
4	795468	1
5	041795	1
6	607251	1
7	091658	1
8	785859	1
9	790210	1
10	142903	1
11	483306	1
12	251893	1
13	094396	1
14	135750	1
15	883954	1
16	109630	1
17	524835	1
18	768962	1
19	141570	1
20	154005	1
21	946984	1
22	334123	1
23	982834	1
24	735101	1
25	631690	1
26	256523	1
27	641081	1
28	877597	1
29	163243	1
30	264309	1
31	782787	1
32	276639	1
33	471384	1
34	097750	1
35	148593	1
36	269750	1
37	562409	1
38	702410	1
39	696901	1
40	364953	1
41	296047	1
42	708537	1
43	150131	1
44	686237	1
45	349142	1
46	248187	1
47	709137	1
48	145490	1
49	663896	1
50	766142	1

Table 4.3.11: *6-digit combinations at 0:31+ generated by DIGIPASS 3 – provided by DNB.*

1	212618	1
2	344110	1
3	252937	1
4	675578	1
5	286695	1
6	202030	1
7	291264	1
8	662409	1
9	819587	1
10	442472	1
11	286419	1
12	656461	1
13	633494	1
14	725070	1
15	127319	1
16	608255	1
17	258728	1
18	250468	1
19	390829	1
20	037612	1
21	368935	1
22	329899	1
23	828425	1
24	469870	1
25	885122	1
26	245701	1
27	447492	1
28	212712	1
29	103473	1
30	759738	1
31	591590	1
32	070855	1
33	519644	1
34	204218	1
35	594627	1
36	544390	1
37	702021	1
38	889430	1
39	621900	1
40	936969	1
41	555510	1
42	710395	1
43	160084	1
44	706565	1
45	127936	1
46	433548	1
47	856022	1
48	763254	1
49	020998	1
50	934562	1

Table 4.3.12: *6-digit combinations at 0:35+ generated by DIGIPASS 3 – provided by DNB.*

1	213291	1
2	708328	1
3	307836	1
4	884696	1
5	441123	1
6	202461	1
7	847966	1
8	156151	1
9	619361	1
10	046973	1
11	698611	1
12	902504	1
13	771174	1
14	793483	1
15	410189	1
16	320058	1
17	503975	1
18	898266	1
19	114131	1
20	751209	1
21	018755	1
22	824362	1
23	761796	1
24	194391	1
25	323962	1
26	291195	1
27	882679	1
28	609798	1
29	403963	1
30	652717	1
31	272932	1
32	197234	1
33	479373	1
34	522408	1
35	734044	1
36	871040	1
37	467660	1
38	007506	1
39	654544	1
40	567902	1
41	418336	1
42	538236	1
43	839992	1
44	825042	1
45	973669	1
46	793805	1
47	105271	1
48	442696	1
49	838488	1
50	811897	1

Table 4.3.13: *6-digit combinations at 0:40+ generated by DIGIPASS 3 – provided by DNB.*

1	409100	1
2	054860	1
3	992861	1
4	661131	1
5	383176	1
6	646742	1
7	816648	1
8	360196	1
9	290302	1
10	877167	1
11	182960	1
12	956250	1
13	144365	1
14	405128	1
15	677674	1
16	774318	1
17	222168	1
18	329304	1
19	821028	1
20	810288	1
21	259120	1
22	595256	1
23	905611	1
24	875893	1
25	851460	1
26	237824	1
27	919863	1
28	599522	1
29	472622	1
30	938818	1
31	445026	1
32	931556	1
33	481475	1
34	559274	1
35	859227	1
36	686084	1
37	571850	1
38	638057	1
39	807869	1
40	469384	1
41	481839	1
42	255190	1
43	915895	1
44	008298	1
45	601014	1
46	133098	1
47	012327	1
48	120513	1
49	299086	1
50	459683	1

Table 4.3.14: *6-digit combinations at 1:00+ generated by DIGIPASS 3 – provided by DNB.*

1	655701	1
2	005201	1
3	885504	1
4	057147	1
5	855452	1
6	590513	1
7	098801	1
8	608530	1
9	136557	1
10	043437	1
11	177660	1
12	241395	1
13	101858	1
14	984255	1
15	808722	1
16	383066	1
17	807342	1
18	536205	1
19	229046	1
20	559382	1
21	471823	1
22	424549	1
23	066871	1
24	768840	1
25	002321	1
26	223792	1
27	461004	1
28	960323	1
29	525156	1
30	806403	1
31	511028	1
32	191811	1
33	562269	1
34	105078	1
35	887353	1
36	485461	1
37	684902	1
38	775147	1
39	239838	1
40	910459	1
41	613966	1
42	705239	1
43	535274	1
44	461550	1
45	138738	1
46	974819	1
47	423778	1
48	747819	1
49	301186	1
50	633207	1

Table 4.3.15: *6-digit combinations at 1:04+ generated by DIGIPASS 3 – provided by DNB.*

1	458452	1
2	057098	1
3	608460	1
4	901705	1
5	051952	1
6	036834	1
7	496552	1
8	092849	1
9	170299	1
10	200486	1
11	419756	1
12	957156	1
13	883229	1
14	311357	1
15	519821	1
16	972349	1
17	056844	1
18	881423	1
19	415236	1
20	708109	1
21	009729	1
22	109302	1
23	958746	1
24	414163	1
25	462460	1
26	840877	1
27	391621	1
28	627759	1
29	208374	1
30	359387	1
31	842359	1
32	333739	1
33	044326	1
34	132204	1
35	902569	1
36	433094	1
37	821542	1
38	827018	1
39	455284	1
40	821422	1
41	846671	1
42	430689	1
43	037472	1
44	686698	1
45	911552	1
46	592222	1
47	367251	1
48	881286	1
49	938897	1
50	387690	1

Table 4.3.16: *6-digit combinations at 1:19+ generated by DIGIPASS 3 – provided by DNB.*

1	279810	1
2	614836	1
3	770428	1
4	408420	1
5	326926	1
6	519285	1
7	773241	1
8	977074	1
9	988485	1
10	306825	1
11	446020	1
12	383746	1
13	778638	1
14	909111	1
15	302223	1
16	446608	1
17	511447	1
18	027053	1
19	918418	1
20	173908	1
21	798266	1
22	848224	1
23	964450	1
24	212399	1
25	146411	1
26	893034	1
27	080324	1
28	024260	1
29	475839	1
30	588271	1
31	547155	1
32	923889	1
33	028682	1
34	366665	1
35	771000	1
36	315112	1
37	620326	1
38	418725	1
39	108358	1
40	739941	1
41	754517	1
42	229204	1
43	743157	1
44	176010	1
45	514411	1
46	471432	1
47	569982	1
48	624124	1
49	931619	1
50	895670	1

Table 4.3.17: *6-digit combinations at 1:21+ generated by DIGIPASS 3 – provided by DNB.*

1	813947	1
2	812875	1
3	329607	1
4	074179	1
5	037184	1
6	264883	1
7	819245	1
8	117610	1
9	412117	1
10	147084	1
11	481320	1
12	976368	1
13	299745	1
14	725801	1
15	232602	1
16	761932	1
17	159625	1
18	042388	1
19	777649	1
20	360394	1
21	945300	1
22	766750	1
23	210064	1
24	902798	1
25	572519	1
26	619281	1
27	030556	1
28	571411	1
29	055694	1
30	261528	1
31	965079	1
32	666838	1
33	580324	1
34	526412	1
35	804642	1
36	862686	1

A2 Sparebanken Møre

Table 4.4.6: *6-digit combination at time interval 0:30+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	763323	1
2	763323	0
3	845596	1
4	845596	0
5	901210	1
6	901210	0
7	032401	1
8	032401	0
9	101593	1
10	101593	0
11	232151	1
12	232151	0
13	344425	1
14	344425	0
15	445382	1
16	445382	0
17	525330	1
18	525330	0
19	651427	1
20	651427	0
21	741418	1
22	741418	0
23	870134	1
24	870134	0
25	932355	1
26	932355	0
27	002232	1
28	002232	0
29	120415	1
30	120415	0
31	219404	1
32	219404	0
33	366444	1
34	366444	0
35	456146	1
36	456146	0
37	517309	1
38	517309	0
39	631426	1
40	631426	0
41	721066	1
42	721066	0
43	829190	1
44	829190	0
45	975232	1
46	975232	0
47	006532	1
48	006532	0
49	169954	1
50	169954	0

Table 4.4.7: *6-digit combination at time interval 0:40+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	361309	1
2	361309	0
3	475984	1
4	502253	1
5	502253	0
6	643316	1
7	744955	1
8	744955	0
9	872516	1
10	872516	0
11	942442	1
12	035411	1
13	035411	0
14	156665	1
15	292525	1
16	292525	0
17	381111	1
18	381111	0
19	455117	1
20	516022	1
21	516022	0
22	656521	1
23	656521	0
24	705260	1
25	705260	0
26	857802	1
27	904373	1
28	904373	0
29	036822	1
30	101080	1
31	101080	0
32	220563	1
33	220563	0
34	320051	1
35	413523	1
36	413523	0
37	581579	1
38	635304	1
39	635304	0
40	793206	1
41	793206	0
42	890705	1
43	954992	1
44	954992	0
45	016030	1
46	016030	0
47	164310	1
48	200670	1
49	200670	0
50	349441	1

Table 4.4.8: *6-digit combination at time interval 0:45+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	852005	1
2	960557	1
3	026403	1
4	026403	0
5	112215	1
6	215167	1
7	360196	1
8	360196	0
9	418944	1
10	550716	1
11	631417	1
12	631417	0
13	731191	1
14	891203	1
15	891203	0
16	940113	1
17	002440	1
18	002440	0
19	143980	1
20	204285	1
21	204285	0
22	301862	1
23	436410	1
24	511364	1
25	511364	0
26	633204	1
27	712403	1
28	840521	1
29	938818	1
30	938818	0
31	095299	1
32	102488	1
33	250016	1
34	250016	0
35	325904	1
36	404725	1
37	504206	1
38	504206	0
39	696145	1
40	790356	1
41	805566	1
42	805566	0
43	915895	1
44	082927	1
45	082927	0
46	105082	1
47	280494	1
48	341482	1
49	341482	0

Table 4.4.9: *6-digit combination at time interval 0:50+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	943361	1
2	067154	1
3	129534	1
4	129534	0
5	281253	1
6	334113	1
7	453895	1
8	530765	1
9	530765	0
10	670045	1
11	775356	1
12	823323	1
13	913444	1
14	913444	0
15	056507	1
16	102424	1
17	289665	1
18	379731	1
19	379731	0
20	482581	1
21	513163	1
22	603103	1
23	799290	1
24	799290	0
25	845149	1
26	903271	1
27	094693	1
28	126242	1
29	210066	1
30	210066	0
31	320556	1
32	432591	1
33	511552	1
34	691802	1
35	691802	0
36	784555	1
37	849411	1
38	994080	1
39	153543	1
40	153543	0
41	203960	1
42	323745	1
43	404450	1
44	595584	1
45	595584	0

Table 4.4.10: *6-digit combination at time interval 0:55+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	555018	1
2	649548	1
3	752153	1
4	810861	1
5	952145	1
6	065354	1
7	065354	0
8	110280	1
9	215191	1
10	310183	1
11	430321	1
12	532327	1
13	690234	1
14	745720	1
15	853324	1
16	853324	0
17	931248	1
18	050000	1
19	198440	1
20	253182	1
21	301121	1
22	458481	1
23	513945	1
24	513945	0
25	674404	1
26	720715	1
27	851143	1
28	919627	1
29	048434	1
30	184864	1
31	201530	1
32	201530	0
33	360329	1
34	408395	1
35	553322	1
36	622110	1
37	724118	1
38	724118	1
39	880978	1
40	905742	1

Table 4.4.11: *6-digit combination at time interval 0:56+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	365859	1	11	362433	1	21	249235	1
2	401534	1	12	422054	1	22	320723	1
3	535519	1	13	559268	1	23	449300	1
4	615053	1	14	625098	1	24	518953	1
5	704423	1	15	745573	1	25	633029	1
6	830210	1	16	831263	1	26	725891	1
7	942205	1	17	908632	1	27	725891	0
8	073314	1	18	908632	0	28	851912	1
9	073314	0	19	083779	1	29	902212	1
10	203165	1	20	191182	1	30	018188	1

Table 4.4.12: *6-digit combination at time interval 0:59+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	435201	1	11	357820	1	21	225203	1
2	551148	1	12	445101	1	22	385892	1
3	644683	1	13	513428	1	23	457443	1
4	644683	0	14	609333	1	24	543464	1
5	737412	1	15	751840	1	25	658883	1
6	824188	1	16	751840	0	26	786218	1
7	931121	1	17	830314	1	27	850865	1
8	069440	1	18	933529	1	28	951104	1
9	111606	1	19	054175	1	29	951104	0
10	255060	1	20	105183	1	30	042811	1
31	192130	1	41	130040	1			
32	245653	1	42	130040	0			
33	310143	1	43	221853	1			
34	442423	1	44	354432	1			
35	557305	1	45	444950	1			
36	643370	1	46	513855	1			
37	781622	1	47	699205	1			
38	812460	1	48	771421	1			
39	950001	1	49	824026	1			
40	012418	1	50	934751	1			

Table 4.4.13: *6-digit combination at time interval 01:00+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	625237	1
2	752301	1
3	858016	1
4	910094	1
5	026440	1
6	127102	1
7	264276	1
8	301037	1
9	441596	1
10	538873	1
11	637610	1
12	732531	1
13	870393	1
14	924818	1
15	052222	1
16	104132	1
17	250083	1
18	250083	0
19	305030	1
20	403207	1
21	531731	1
22	644963	1
23	706899	1
24	932726	1
25	067729	1
26	142351	1
27	231940	1
28	314570	1
29	480048	1
30	546280	1
31	622435	1
32	622435	0
33	760133	1
34	845993	1
35	998321	1
36	059251	1
37	140263	1
38	252751	1
39	320072	1
40	484386	1

Table 4.4.14: *6-digit combination at time interval 01:01+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	051485	1
2	150930	1
3	200344	1
4	331304	1
5	440154	1
6	584655	1
7	673212	1
8	701346	1
9	836482	1
10	903925	1
11	047094	1
12	152012	1
13	276869	1
14	302821	1
15	440811	1
16	555231	1
17	643412	1
18	643412	0
19	764733	1
20	819125	1
21	960337	1
22	030004	1
23	122287	1
24	200340	1
25	315502	1
26	466303	1
27	562119	1
28	657100	1
29	730144	1
30	831743	1
31	920523	1
32	053577	1
33	125217	1
34	288165	1
35	366024	1
36	465253	1
37	547541	1
38	665725	1
39	784202	1
40	784202	0

Table 4.4.15: *6-digit combination at time interval 01:02+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	961259	1
2	094907	1
3	199133	1
4	275295	1
5	339254	1
6	425601	1
7	580243	1
8	602451	1
9	743745	1
10	804554	1
11	917020	1
12	917020	0
13	099584	1
14	151091	1
15	233145	1
16	361240	1
17	412204	1
18	507955	1
19	631471	1
20	721294	1
21	838015	1
22	975052	1
23	032341	1
24	131131	1
25	244502	1
26	392669	1
27	434700	1
28	543139	1
29	647356	1
30	752022	1
31	815113	1
32	983813	1
33	093392	1
34	113091	1
35	235269	1
36	383170	1
37	402545	1
38	562456	1
39	618420	1
40	711057	1
41	711057	0
42	846211	1
43	970642	1

Table 4.4.16: *6-digit combination at time interval 01:03+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	721336	1
2	863430	1
3	953391	1
4	041012	1
5	172910	1
6	235590	1
7	313200	1
8	452894	1
9	504251	1
10	627504	1
11	701884	1
12	861103	1
13	992105	1
14	094030	1
15	197063	1
16	217251	1
17	316055	1
18	487914	1
19	505533	1
20	633536	1
21	767520	1
22	813762	1
23	996122	1
24	033535	1
25	117383	1
26	294015	1
27	319231	1
28	480564	1
29	555151	1
30	606462	1

31	761420	1
32	891929	1
33	891929	0
34	925528	1
35	054453	1
36	105902	1
37	296960	1
38	331213	1
39	449515	1
40	555795	1
41	645261	1
42	741942	1
43	824134	1
44	942157	1
45	005151	1
46	101754	1
47	200220	1
48	349308	1
49	405305	1
50	521374	1
51	644842	1
52	742312	1
53	851111	1
54	976953	1
55	067613	1
56	182043	1
57	280450	1
58	323375	1
59	483586	1
60	581500	1
61	638035	1
62	711704	1
63	832300	1
64	964052	1
65	014306	1
66	123537	1
67	257430	1
68	350112	1
69	434529	1
70	567781	1
71	602532	1
72	723172	1
73	854140	1
74	993090	1
75	036320	1
76	149501	1
77	252642	1
78	252642	0

Table 4.4.17: *6-digit combination at time interval 01:04+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	746672	1
2	821987	1
3	931692	1
4	015166	1
5	117484	1
6	250232	1
7	359282	1
8	413031	1
9	550352	1
10	632223	1
11	772543	1
12	819045	1
13	954331	1
14	031300	1
15	101470	1
16	202772	1
17	309357	1
18	407534	1
19	523082	1
20	678401	1
21	757501	1
22	818998	1
23	902931	1
24	038209	1
25	109104	1
26	210502	1
27	331091	1
28	410257	1
29	550306	1
30	625280	1
31	735514	1
32	800231	1
33	997888	1
34	051225	1
35	157053	1
36	208884	1
37	362541	1
38	482369	1
39	515785	1
40	642403	1
41	778434	1
42	818452	1
43	938207	1
44	025321	1
45	151093	1
46	281421	1
47	347302	1
48	456254	1
49	504550	1
50	603536	1
51	775206	1
52	802522	1
53	945155	1
54	032413	1
55	142385	1
56	212436	1
57	305957	1
58	427590	1
59	532423	1
60	625685	1

61	710160	1
62	808006	1
63	920025	1
64	013802	1
65	113203	1
66	204325	1
67	355728	1
68	451383	1
69	552922	1
70	699155	1
71	725322	1
72	868012	1
73	931522	1
74	030705	1
75	178331	1
76	284265	1
77	328050	1
78	417453	1
79	542362	1
80	674800	1
81	745582	1

Table 4.4.18: *6-digit combination at time interval 01:05+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	919043	1
2	051399	1
3	125538	1
4	205535	1
5	324883	1
6	474516	1
7	519245	1
8	620712	1
9	719616	1
10	840225	1
11	929412	1
12	002367	1
13	185420	1
14	232086	1
15	345401	1
16	473454	1
17	590572	1
18	674032	1
19	705183	1
20	859538	1
21	919713	1
22	055523	1
23	122333	1
24	245052	1
25	345415	1
26	444116	1
27	556941	1
28	674322	1
29	716307	1
30	823641	1
31	912257	1
32	084542	1
33	143252	1
34	270311	1
35	342325	1
36	426383	1
37	546791	1
38	649286	1
39	753011	1
40	842223	1

Table 4.4.19: *6-digit combination at time interval 01:20+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	931251	1
2	051399	1
3	125538	1
4	205535	1
5	324883	1
6	474516	1
7	519245	1
8	620712	1
9	719616	1
10	840225	1
11	929412	1
12	002367	1
13	185420	1
14	232086	1
15	345401	1
16	473454	1
17	590572	1
18	674032	1
19	705183	1
20	859538	1

Table 4.4.20: *6-digit combination at time interval 01:30+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	006323	1
2	149304	1
3	300874	2
4	412247	1
5	634925	2
6	703117	1
7	833140	1
8	002504	2
9	155191	1
10	359429	2
11	422612	1
12	609132	2
13	783300	1
14	896804	1
15	114285	2
16	307338	2
17	400332	1
18	604410	2
19	741322	1
20	823335	1

Table 4.4.21: *6-digit combination at time interval 02:00+ DIGIPASS 3 – provided by Sparebanken Møre.*

1	027880	1
2	261172	2
3	448418	2
4	654174	2
5	791135	1
6	923019	2
7	153452	2
8	320234	2
9	522752	2
10	799232	2
11	932647	2
12	115583	2
13	245010	1
14	451125	2
15	611044	2
16	844712	2
17	088305	2
18	257639	2
19	415532	2
20	647210	2
21	748385	1
22	917334	2
23	126690	2
24	357451	2
25	537342	2
26	744212	2
27	901241	2
28	113034	2

Table 4.4.22: *6-digit combination at time interval 05:00+* DIGIPASS 3 – provided by Sparebanken Møre.

1	239816	1
2	735359	5
3	190567	4
4	635045	5
5	143224	5
6	572632	4
7	047427	5
8	590316	5
9	910388	4
10	443315	5
11	950482	5
12	343587	4
13	843372	5

A3 Sparebank 1 Nordvest

Table 4.5.6: *6-digit combination at time interval 0:29+ – token provided by Sparebank 1 Nordvest.*

1	949231	1
2	943887	0
3	916614	1
4	956587	1
5	939699	1
6	938677	0
7	988474	1
8	922805	1
9	999131	1
10	936142	1
11	921632	1
12	985529	1
13	939087	1
14	906001	1
15	901201	0
16	975156	1
17	938204	1
18	989291	1
19	982973	0
20	940510	1
21	931607	1
22	909416	1
23	983344	1
24	918208	1
25	919829	0
26	920328	1
27	941728	1
28	931987	1
29	972092	1
30	976798	0
31	994274	1
32	906110	1
33	948295	1
34	902120	1
35	930164	1
36	963860	1
37	996849	1
38	953965	1
39	963014	1
40	905627	1
41	969831	1
42	963113	0
43	921307	1
44	909926	1
45	931936	1
46	913937	1
47	950454	1
48	902715	1
49	993848	1
50	998309	0

Table 4.5.7: *6-digit combination at time interval 0:30+ – token provided by Sparebank 1 Nordvest.*

1	935424	1
2	929666	1
3	945478	1
4	980643	1
5	937571	1
6	967883	1
7	970416	1
8	979435	0
9	995915	1
10	939891	1
11	922658	1
12	994820	1
13	939185	1
14	957986	1
15	977784	1
16	973098	0
17	959116	1
18	933000	1
19	965966	1
20	985154	1
21	905970	1
22	928029	1
23	917040	1
24	910562	0
25	936359	1
26	990019	1
27	962996	1
28	950199	1
29	968871	1
30	976229	1

31	967420	1
32	908778	1
33	906273	0
34	909553	0
35	977496	1
36	989556	1
37	985604	0
38	948307	1
39	910048	1
40	978284	1
41	958723	1
42	935197	1
43	970616	1
44	968025	1
45	956948	1
46	946990	1
47	915503	1
48	930815	1
49	971457	1
50	972310	0

Table 4.5.8: *6-digit combination at time interval 0:31+ – token provided by Sparebank 1 Nordvest.*

1	935055	1
2	931142	0
3	969709	1
4	993181	1
5	967086	1
6	928030	1
7	923644	0
8	903566	1
9	951817	1
10	966597	1
11	940336	1
12	975549	1
13	987509	1
14	971272	1
15	935541	1
16	937528	0
17	929372	1
18	978050	1
19	930824	1
20	970301	1
21	912860	1
22	924341	1
23	975163	1
24	902252	1
25	963434	1
26	916070	1
27	916854	0

Table 4.5.9: *6-digit combination at time interval 0:35+ – token provided by Sparebank 1 Nordvest.*

1	967109	1
2	946616	1
3	938591	1
4	944889	1
5	989870	1
6	974460	1
7	939687	1
8	913842	1
9	930808	1
10	983806	1
11	919877	1
12	979351	1
13	981852	1
14	933751	1
15	903263	1
16	953896	1
17	932918	1
18	909710	1
19	972601	1
20	908764	1
21	937784	1
22	974469	1
23	913746	1
24	943427	1
25	939991	1
26	942816	1
27	911421	1
28	925737	1
29	976329	1
30	953091	1
31	989636	1
32	973472	1
33	914224	1
34	911761	0
35	919127	0
36	986963	1
37	913706	1
38	949761	1
39	946991	0
40	978765	1
41	910891	1
42	924069	1
43	924743	0
44	989446	1
45	920329	1
46	907355	1
47	937633	1
48	932442	0
49	919518	1
50	939056	1

Table 4.5.10: *6-digit combination at time interval 0:40+ – token provided by Sparebank Nordvest.*

1	916714	1
2	943288	1
3	927427	1
4	932288	1
5	920902	1
6	951520	1
7	925839	1
8	360196	1
9	997470	1
10	926368	1
11	975796	1
12	956964	1
13	977421	1
14	909234	1
15	931693	1
16	944134	1
17	906527	1
18	928538	1
19	949489	1
20	977180	1
21	963356	1
22	953296	1
23	999762	1
24	937175	1
25	943846	1
26	928129	1
27	990118	1
28	949789	1
29	942252	0
30	915723	1
31	905299	1
32	936597	1
33	903804	1
34	961897	1
35	958775	1
36	981094	1
37	929989	1
38	900619	1
39	938475	1
40	942266	1
41	969610	1
42	948861	1
43	957669	1
44	946873	1
45	983271	1
46	966309	1
47	961822	0
48	995442	1
49	967886	1
50	921055	1

Table 4.5.11: *6-digit combination at time interval 01:00+ – token provided by Sparebank 1 Nordvest.*

1	998982	1
2	979676	1
3	948757	1
4	921550	1
5	922335	0
6	936608	1
7	900363	1
8	972688	1
9	972146	0
10	913143	1
11	970622	1
12	945776	1
13	945575	0
14	983580	1
15	954372	1
16	948822	1
17	907959	1
18	900726	0
19	912771	1
20	903660	1
21	946296	1
22	947017	0
23	927041	1
24	969729	1
25	915106	1
26	948976	1
27	945014	0
28	992215	1
29	958371	1
30	914516	1
31	964470	1
32	943044	1
33	949643	0
34	932190	1
35	961743	1
36	922903	1
37	953656	1
38	980661	1
39	977125	1
40	975529	0
41	951117	1
42	938161	1
43	963724	1
44	969434	0
45	945820	1
46	960417	1
47	934420	1
48	936295	0
49	978621	1
50	978697	0

Table 4.5.12: *6-digit combination at time interval 01:29+ – token provided by Sparebank 1 Nordvest.*

1	998920	1
2	993937	0
3	927604	1
4	917881	1
5	912623	0
6	903278	1
7	905395	0
8	961700	1
9	962126	0
10	925175	1
11	944711	1
12	940741	0
13	987848	1
14	960433	1
15	923936	1
16	937515	1
17	935270	0
18	984097	1
19	972025	1
20	926505	1
21	940525	1
22	942248	0
23	924351	1
24	941529	1
25	913902	1
26	906422	1
27	943600	1
28	976422	1
29	900163	1
30	921606	1
31	991761	1
32	913284	1
33	978419	1
34	961476	1
35	969258	0
36	950751	1
37	918754	1
38	977634	1
39	969832	1
40	950178	1
41	994200	1
42	961853	1
43	975442	1
44	984978	1
45	933867	1
46	904114	1
47	903768	0
48	967995	1
49	917520	1
50	996836	1

Table 4.5.13: *6-digit combination at time interval 01:30+ – token provided by Sparebank 1 Nordvest.*

1	947986	1
2	921023	1
3	957696	1
4	955120	0
5	957678	0
6	962020	1
7	976724	1
8	969650	1
9	965730	0
10	945699	1
11	983197	1
12	970971	1
13	938338	1
14	963685	1
15	994757	1
16	998215	0
17	922349	1
18	946803	1
19	989686	1
20	987288	0
21	916522	1
22	982320	1
23	901754	1
24	953071	1
25	900106	1
26	915766	1
27	916157	0
28	967332	1
29	944608	1
30	939714	1
31	920155	1
32	945998	1
33	968654	1
34	991259	1
35	915580	1
36	920598	1
37	903432	1
38	934470	1
39	974650	1
40	933501	1
41	981989	1
42	927066	1
43	901241	1
44	951825	1
45	908238	1
46	993572	1
47	904554	1
48	972003	1
49	914184	1
50	908985	1

Table 4.5.14: *6-digit combination at time interval 02:00+ – token provided by Sparebank 1 Nordvest.*

1	949147	1
2	948767	0
3	908650	1
4	915862	1
5	944676	1
6	915337	1
7	996812	1
8	980470	1
9	983933	0
10	980748	0
11	918324	1
12	990998	1
13	915721	1
14	918750	0
15	984755	1
16	981721	0
17	907061	1
18	915415	1
19	971212	1
20	989488	1
21	936207	1
22	986719	1
23	901128	1
24	940692	1
25	923371	1
26	975347	1
27	955509	1
28	948193	1
29	909602	1
30	958454	1

Table 4.5.15: *6-digit combination at time interval 05:00+ – token provided by Sparebank 1 Nordvest.*

1	907552	1
2	938650	1
3	957852	1
4	914341	1
5	940053	1
6	951007	1
7	998842	1
8	948238	1
9	954743	1
10	968631	1
11	914320	1
12	956473	1
13	978429	1
14	983638	1
15	908604	1
16	919501	1
17	938409	1
18	957263	1
19	982835	1
20	951198	1
21	959052	1
22	934787	1
23	984480	1
24	960044	1
25	934965	1
26	948184	1

A4 Nordea

Table 4.6.6: *8-digit combination at time interval 0:15+ – token provided by Nordea.*

1	42274042	1
2	43974240	0
3	44042429	0
4	45469128	0
5	46443594	0
6	47584490	0
7	48185415	0
8	49821604	0
9	50549357	1
10	51336103	0
11	52192000	0
12	53337567	0
13	54413525	0
14	55490843	0
15	56433803	0
16	57426205	0
17	58103525	0
18	69518361	1
19	60122393	0
20	61123221	0
21	63066250	0
22	64034472	0
23	65317839	0
24	66893233	0
25	77293854	1
26	78103357	0
27	79113800	0
28	70531083	0
29	71832701	0
30	72005112	0

31	73034096	0
32	74534960	0
33	75242158	0
34	76142001	0
35	87811264	1
36	88103521	0
37	89420252	0
38	81925518	0
39	82858355	0
40	83902505	0
41	84556217	0
42	85031736	0
43	86833605	0
44	87534496	0
45	88545032	0
46	99045549	1
47	92355151	0
48	93654334	0
49	94057340	0

Table 4.6.7: *8-digit combination at time interval 0:25+ – token provided by Nordea.*

1	97551256	1
2	98461427	0
3	09551225	1
4	00942870	0
5	01334475	0
6	02583513	0
7	13571311	1
8	14083334	0
9	15706297	0
10	16070537	0
11	17242401	0
12	28201620	1
13	29258595	0
14	20514239	0
15	21415031	0
16	22004573	0
17	33632395	1
18	34715068	0
19	35052151	0
20	36320453	0
21	37816752	0
22	48495015	1
23	49563414	0
24	40350314	0
25	41131163	0
26	42550297	0
27	53440319	1
28	54953404	0
29	55418102	0
30	56746692	0
31	57101097	0
32	58041802	0
33	69003422	1
34	60954073	0
35	61217524	0
36	62919706	0
37	63917204	0
38	74411032	1
39	75852611	0
40	76938365	0
41	77415503	0
42	78038494	0
43	89394952	1
44	80424335	0
45	81576638	0
46	82117345	0
47	83563761	0
48	94586555	1
49	95541081	0

Table 4.6.8: *8-digit combination at time interval 0:40+ – token provided by Nordea.*

1	92500131	1
2	93217221	0
3	04212511	1
4	05131036	0
5	06823245	0
6	17151155	1
7	18519571	0
8	19201740	0
9	20304362	1
10	21925571	0
11	22865742	0
12	33602201	1
13	34741844	0
14	35155132	0
15	46274139	1
16	47911586	0
17	48852602	0
18	59051202	1
19	50131191	0
20	51653031	0
21	62282296	1
22	63469520	0
23	64597418	0
24	75047172	1
25	76262633	0
26	77325822	0
27	78245320	0
28	89105774	1
29	80010500	0
30	81649041	0
31	92403012	1
32	93553529	0
33	94045455	0
34	05052221	1
35	06882357	0
36	07230554	0
37	18163211	1
38	19255328	0
39	10403517	0
40	21750544	1
41	22135105	0
42	23245333	0
43	34301613	1
44	35807134	0
45	36224419	0
46	47206497	1
47	48266804	0
48	49917041	0
49	50716944	1
50	51329408	0

Table 4.6.9: *8-digit combination at time interval 0:55+ – token provided by Nordea.*

1	78642232	1
2	79225849	0
3	80342380	1
4	91979605	1
5	92224725	0
6	93512032	0
7	04354136	1
8	05405406	0
9	16558333	1
10	17346803	0
11	28203103	1
12	29567917	0
13	20730454	0
14	31544032	1
15	32141521	0
16	43056824	1
17	44239362	0
18	55481303	1
19	56572150	0
20	67408103	1
21	68085832	0
22	69098430	0
23	70331130	1
24	71913503	0
25	82134280	1
26	83438296	0
27	94300602	1
28	95200008	0
29	06297201	1
30	07004463	0
31	08703429	0
32	19552725	1
33	10717434	0
34	21125153	1
35	22105930	0
36	23601784	0
37	34401212	1
38	35084596	0
39	46798759	1
40	47555450	0
41	58182484	1
42	59962991	0
43	60183048	1
44	61171748	0
45	62575019	0
46	73214176	1
47	74903844	0
48	85245871	1
49	86766354	0

Table 4.6.10: *8-digit combination at time interval 0:59+ – token provided by Nordea.*

1	97050101	1
2	98153194	0
3	09118925	1
4	00992795	0
5	11804420	1
6	12510355	0
7	23870150	1
8	25702820	0
9	36554356	1
10	48031615	1
11	49579053	0
12	50110505	1
13	51958924	0
14	62293551	1
15	63221549	0
16	64454064	0
17	75613700	1
18	76215930	0
19	87432946	1
20	88304661	0
21	99770421	1
22	90917245	0
23	01048417	1
24	02507047	0
25	13234843	1
26	14226460	0
27	25011186	1
28	26143011	0
29	37544823	1
30	38444263	0

Table 4.6.11: *8-digit combination at time interval 01:00+ – token provided by Nordea.*

1	44411841	1
2	45331533	0
3	56205817	1
4	57702786	0
5	68216725	1
6	69542400	0
7	70163645	1
8	71352285	0
9	82957883	1
10	83260946	0
11	94606299	1
12	95398751	0
13	06213101	1
14	07476842	0
15	18405738	1
16	19441594	0
17	20151622	1
18	21421114	0
19	32011046	1
20	33440127	0
21	44623154	1
22	45907011	0
23	56245712	1
24	57140222	0
25	68352177	1
26	69316491	0
27	60289321	0
28	71091503	1
29	72182390	0
30	83134354	1
31	84914675	0
32	95134531	1
33	96337160	0
34	07498803	1
35	08001111	0
36	19538156	1
37	10043410	0
38	21682040	1
39	22554345	0
40	33151222	1
41	34388323	0
42	45034803	1
43	46010508	0
44	47789334	0
45	58950514	1
46	59954540	0
47	60615633	1
48	61478040	0
49	72094574	1
50	73727454	0

Table 4.6.12: *8-digit combination at time interval 01:15+ – token provided by Nordea.*

1	96221440	1
2	07095946	1
3	08073230	0
4	19951571	1
5	10031858	0
6	21872334	1
7	32486313	1
8	33341540	0
9	44934739	1
10	45332402	0
11	56017200	1
12	67247523	1
13	68534145	0
14	79432134	1
15	70362201	0
16	81211466	1
17	92254205	1
18	93014430	0
19	04314150	1
20	05485339	0
21	16120984	1
22	27598431	1
23	28673607	0
24	39533398	1
25	30223313	0
26	41395518	1
27	42651219	0
28	53115390	1
29	64795010	1
30	65430314	0
31	76145918	1
32	77222421	0
33	88237442	1
34	99156163	1
35	90473508	0
36	01022234	1
37	02654558	0
38	13009975	1
39	14706510	0
40	25236083	1
41	36160725	1
42	37924106	0
43	48108587	1
44	49963988	0
45	50225450	1
46	61355196	1
47	62601023	0
48	73550442	1
49	74502445	0
50	85202945	1

Table 4.6.13: *8-digit combination at time interval 01:25+ – token provided by Nordea.*

1	76120589	1
2	77917430	0
3	88214221	1
4	99202514	1
5	90403494	0
6	01750174	1
7	12335225	1
8	23368944	1
9	24823554	0
10	35570349	1
11	46440205	1
12	47102155	0
13	58962831	1
14	69154728	1
15	70553541	1
16	71033874	0
17	82052437	1
18	93806004	1
19	94805511	0
20	05493034	1
21	16799443	1
22	27542112	1
23	28443364	0
24	39736051	1
25	40528551	1
26	41750045	0
27	52881908	1
28	63697601	1
29	74451527	1
30	75355129	0
31	86024224	1
32	97034707	1
33	98645001	0
34	09543294	1
35	10230532	1
36	21510010	1
37	22343732	0
38	33800958	1
39	44282584	1
40	45615279	0
41	56063886	1
42	67656012	1
43	78759423	1
44	79020430	0
45	80107205	1
46	91425571	1
47	92415931	0
48	03172614	1
49	14573129	1
50	25121827	1

Table 4.6.14: *8-digit combination at time interval 01:30+ – token provided by Nordea.*

1	06755010	1
2	07547523	0
3	18213551	1
4	29053429	1
5	30959753	1
6	31037737	0
7	42105258	1
8	53680545	1
9	64021107	1
10	65276843	0
11	76948454	1
12	87174387	1
13	98625765	1
14	99690431	0
15	00002226	1
16	11925436	1
17	22641329	1
18	23700549	0
19	34701862	1
20	45134503	1
21	56355140	1
22	57843017	0
23	68375244	1
24	79835018	1
25	80423915	1
26	81474806	0
27	92521771	1
28	03115033	1
29	14230751	1
30	15711924	0
31	26601196	1
32	37225124	1
33	48061304	1
34	49106352	0
35	50321235	1
36	61215515	1
37	72357434	1
38	73213955	0
39	85315736	1
40	96422266	1
41	07025813	1
42	08610168	0
43	19253354	1
44	20717916	1
45	31361144	1
46	32939314	0
47	43651220	1
48	54212243	1
49	65271080	1
50	66341139	0

Table 4.6.15: *8-digit combination at time interval 01:59+ – token provided by Nordea.*

1	83593001	1
2	74588515	1
3	85200353	1
4	96035458	1
5	07944404	1
6	18446416	1
7	29219405	1
8	30044213	1
9	41552421	1
10	52022177	1
11	63252784	1
12	74525723	1
13	85525423	1
14	86544537	0
15	97146050	1
16	08431383	1
17	19651650	1
18	20110202	1
19	31016144	1
20	42791743	1
21	53241655	1
22	64039325	1
23	75710179	1
24	86935234	1
25	97314575	1
26	08463380	1
27	19097350	1
28	20129564	1
29	31552034	1
30	42065765	1

Table 4.6.16: *8-digit combination at time interval 02:00+ – token provided by Nordea.*

1	56000733	1
2	67706300	1
3	78682232	1
4	89821115	1
5	90253382	1
6	01348142	1
7	12518348	1
8	23425020	1
9	34033236	1
10	45353292	1
11	56168695	1
12	67283752	1
13	78274951	1
14	89922322	1
15	90472650	1
16	01962153	1
17	12287113	1
18	23531538	1
19	34710452	1
20	45037122	1
21	56325121	1
22	67745501	1
23	78930234	1
24	89780403	1
25	90324758	1
26	01563910	1
27	12432441	1
28	23448474	1
29	34493980	1
30	45025535	1
31	56232120	1
32	67711540	1
33	78602816	1
34	89203001	1
35	90285529	1
36	01441433	1
37	12486291	1
38	23850203	1
39	34975111	1
40	45722440	1
41	56465440	1
42	67465503	1
43	78536889	1
44	89997834	1
45	90045514	1
46	01902323	1
47	12367411	1
48	23078179	1
49	34653732	1
50	45331290	1

Table 4.6.17: *8-digit combination at time interval 05:00+ – token provided by Nordea.*

1	62043714	1
2	83020057	2
3	04457543	2
4	35352379	3
5	56505294	2
6	77903773	2
7	08529576	3
8	29154540	2
9	50522003	3
10	71378348	2

Table 4.6.18: *8-digit combination at time interval 06:00+ – token provided by Nordea.*

1	09302658	1
2	30703608	3
3	51230586	2
4	82158685	3
5	13353265	3
6	44835012	3
7	75473764	3
8	96264891	2
9	17081161	2
10	48814556	3

Table 4.6.19: *8-digit combination at time interval 08:00+ – token provided by Nordea.*

1	89585918	
2	20301545	4
3	61707880	4
4	02236084	4
5	43496525	4
6	84132906	4
7	25520707	4
8	66017064	4
9	07460226	4
10	48961218	4

Table 4.6.20: *8-digit combination at time interval 10:00+ – token provided by Nordea.*

1	88430800	4
2	29302475	4
3	60411388	4
4	01465415	4
5	42023669	4
6	83075210	4
7	24434338	4
8	65211604	4
9	06362843	4
10	47230874	4

Bibliography

[1] Verizon, "Data Breach Investigations Report," Verizon, New York City, 2020. [Online]. Available: https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf

[2] I. A. Semaev, "Experimental Study of DIGIPASS GO3 and the Security of Authentication," Univeristy of Bergen, Bergen, 2015. [Online]. Available: https://eprint.iacr.org/2015/609.pdf

[3] S. T. Kent and L. I. Millett, "Authentication in the Abstract," in *Who Goes There?: Authentication Through the Lens of Privacy*. Washington, DC: The National Academies Press, 2003, pp. 33-54.

[4] S. Tumin and S. Encheva, "A Closer Look at Authentication and Authorization Mechanisms for Web-based Applications," *Applied Information Science*. Accessed: Jan. 15, 2020, ISBN: 978-1-61804-089-3

[5] R. K. Rainer Jr., B. Prince and C. Cegielski, "Infirmation Security," in *Introduction to Information Systems*, 5th ed., Singapore: Wiley, 2015, pp. 72-99.

[6] A. Vapen, "Background," in *Web Authentication using Third-Parties in Untrusted Environments*, Linköping, Sweden: Linköping University, 2016, pp. 9-28.

[7] Thales, "Biometrics: authentication & identification (definition, trends, use cases, laws and latest news) - 2020 review," Accessed: Okt. 23, 2019. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government /inspired/biometrics

[8] G. Paul and J. Irvine, "Fingerprint Authentication is here, but are we ready for what it brings?," University of Strathclyde: Department of Electronic & Electrical Engineering, Glasgow, United Kingdom. [Online]. Available: (accessed Nov. 25, 2019).

[9] J. N. Pato and L. I. Millett, "Introduction and Fundamental Concepts," in *Biometric Recognition: Challenges and Opportunities*, Washington, DC, The National Academies Press, 2010, pp. 15-52.

[10] R. Jiang, S. Al-maadeed, A. Bouridane, D. Crookes and A. Beghdadi, “Recognition og 3D Faces with Missing Parts Based om SIFT and LBP Metods,” in *Biometric Security and Privacy: Opportunities & Challenges in The Big Data Era*, Cham, Switzerland: Springer, 2017, pp. 273-298.

[11] J. M. Kizza, Authentication, “Cyber Crimes and Hacers,” in *A Guide to Computer Network Security*, London : Springer, 2009, pp. 207-226.

[12] Check point research, “Cyberattack trends analysis. Key insights to gear ut for in 2019,” Check point research Software Technologies Ltd, (Vol. 1), 2019. Accessed: Des. 03, 2019. [Online]. Available: http://snt.hr/boxcontent/CheckPointSecurityReport2019_vol01.pdf?fbclid=IwAR1LhXTR4oP4sngbxmnjtrv2EGTQBkgfB6q7KeJPxE609KlwO0mxaBtqglY

[13] The Norwegian Tax Administration. “Norwegian national identity number.” skatteetaten.no. https://www.skatteetaten.no/en/person/nationalregistry/birthandname-selection/children-born-in-norway/national-id-number/ (accessed Des. 15, 2019)

[14] The Norwegian Tax Administration. “D number.” skatteetaten.no. https://www.skatteetaten.no/en/person/foreign/norwegian-identification-number/d-number/?fbclid=IwAR2UCnONuK2sEh4zBEblEWrojD8D5NBOzqwI-he5VicA1s-95hoG8HSxexI (accessed: Des. 15, 2019).

[15] Norwegian Digitalisation Agency. Digir.no. https://www.digdir.no/ (accessed: Des. 19, 2019).

[16] Altinn. “Security levels.” Altinn.no. https://www.altinn.no/en/help/logging-in/miscellaneous-about-logging-in/sikkerhetsniva/ (accessed: Des. 28, 2019).

[17] BankID. “Om oss.” Bankid.no. https://www.bankid.no/privat/om-oss/ (accessed: Jan. 04, 2020).

[18] The Norwegian Digitalisation Agency. “How to order Buypass ID on smart card?” Eid.difi.no. https://eid.difi.no/en/buypass-id/how-order-buypass-id-smart-card (accessed: Jan. 04, 2020).

[19] M. O. Rayes, "One-time password," in *Encyclopedia of Cryptography and Security*, 2nd ed., H. C. A. v. Tilborg and S. Jajodia, Eds., Boston, Springer, 2011, p. 855.

[20] J. J. Stapleton. “Authentication,” in *Security without Obscurity: A Guide to Confidentiality, Authentication, and Integrity*, Boca Raton, CRC Press, 2014, pp. 91-168.

[21] C. Mulliner, R. Borgaonkar, P. Stewin and J.-P. Seifert, "SMS-Based One-Time Passwords: Attacks and Defense," Northeastern University and Technische Universität

Berlin, 2013. Accessed: Jan. 20, 2019. [Online]. Available: https://www.mulliner.org/collin/publications/mulliner_dimva2013.pdf

[22] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache and R. O., "HOTP: An HMAC-Based One-Time Password Algorithm," The Internet Society, 2005. Accessed: Jan. 21, 2020. [Online]. Available: https://www.ietf.org/rfc/rfc4226.txt

[23] D. M'Raihi, S. Bajaj, S. Machani, D. Naccache and J. Rydell, " TOTP: Time-Based One-Time Password Algorithm," 2011 Accessed: Jan. 27, 2020. [Online]. Available: https://www.ietf.org/rfc/rfc6238.txt

[24] D. M'Raihi, S. Machani, M. Pei and J. Rydell, "OCRA: OATH Challenge-Response Algorithm," 2011. Accessed: Feb. 03, 2020. [Online]. Available: https://www.ietf.org/rfc/rfc6287.txt

[25] R. D. Stinson and M. Paterson, Introduction to Cryptography, in *Cryptography: Theory and Practice*, 4th ed. Boca Raton CRC Press, 2019, pp. 1-14.

[26] R. D. Stinson and M. Paterson, Hash Function and message Authentication, in *Cryptography: Theory and Practice*, 4th ed. Boca Raton CRC Press, 2019, pp. 137-184.

[27] D. R. PATEL, "Data Integrity: Cryptographic Hash Functions," in *INFORMATION SECURITY: Theory and Practice*, New Delhi, PHI Learning Pvt. Ltd, 2008.

[28] W. Mehuron, "The Keyed-Hash Message Authentication Code (HMAC)," in *Information Technology Laboratory*, National Institute of Standards and Technology, Gaithersburg, 2002. Accessed: Feb. 20, 2020. [Online]. Available: https://csrc.nist.gov/csrc/media/publications/fips/198/archive/2002-03-06/documents/fips-198a.pdf?fbclid=IwAR0GtgPlmfws1XZ7W5xbzEqwNtlN7ALVaTbeg_FXU0bMZ6RbJHNUBmVN4WM

[29] Thales "History," Thalesgroup.com. https://www.thalesgroup.com/en/global/group/history (Accessed Feb. 17, 2020).

[30] Thales "Banking Tokens - OTP Authentication and Signature Devices," Thalesgroup.com. https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-banking/tokens?fbclid=IwAR0xNG9Cj3bWpTR-1XHxInv4GgK0lfDwP0OigAg6Eaj3ypCdCW5nxyBd530 (Accessed Feb. 17, 2020).

[31] OneSpan. "News_VASCO integrates strong host authentication into DIGIPASS 250,260 and 270." OneSpan.com. https://www.onespan.com/about/news/newsvasco-integrates-strong-host-authentication-digipass-250260-and-270 (Accessed Jan. 20, 2020).

[32] OneSpan. "OneSpan's solutions have helped prevent billions of dollars in fraud." OneSpan.com. https://www.onespan.com/about (Accessed Jan. 20, 20202).

[33] T. K. Hunt, "Form 10-K/A Onespan Inc.10-K/A [Amend] - Annual report [Section 13 and 15(d), not S-K Item 405]" VASCO, USA, 2001. https://sec.report/Document/0000950137-02-002568/?fbclid=IwAR2C3CUsH-L2y3aAa9hZ4Mtm_1mU5_JYbbWrfVc12IOAFGT7fBznOJez1m8

[34] OneSpan. "One-Button Authenticators." OneSpan.com. https://www.onespan.com/products/one-button (Accessed Jan. 20, 2020).

[35] HID Global. "About HID Global." Hidglobal.com. https://www.hidglobal.com/about (Accessed 24 01 2020).

[36] HID Global. "HID® ActivID® One-Time Password (OTP) Tokens." Hidglobal.com. https://www.hidglobal.com/products/cards-and-credentials/activid/one-time-password-tokens (Accessed Feb. 24, 2020).

[37] HID Global. "HID® ActivID® BlueTrust Token." Hidglobal.com. https://www.hidglobal.com/products/cards-and-credentials/activid/bluetrust-token (Accessed Jan. 24, 2020).

[38] DNB. "About the Group." DNB.no. https://www.dnb.no/en/about-us/about-the-group.html (Accessed Feb. 14, 2020).

[39] SpareBanken Møre. "Våre kjerneverdier." Sbm.no. https://www.sbm.no/om-ossir/om-sparebanken-more/om-oss/om-oss---tillegg/vare-kjerneverdier/401/437/ (Accessed Feb. 24, 2020).

[40] Vasco "VASCO Data Security International, Inc," Vasco the authentication company, (version 1,7), 2015. Accessed: Feb. 24, 2020. [Online]. Available: https://csrc.nist.rip/groups/STM/cmvp/documents/140-1/140sp/140sp2432.pdf?fbclid=IwAR1lUbaMsZIHu10diAt2XgbPNo2ctBKfb-TnwOAcYbs7gAsE3FfDNSJ3vxc

[41] SpareBank 1. "Om oss." Sparebank1.no. https://www.sparebank1.no/nb/bank/om-oss.html (Accessed Mar. 20, 2020).

[42] SpareBank 1 Nordvest. "Om oss." Sparebank1.no. https://www.sparebank1.no/nb/nordvest/om-oss.html (Accessed Mar. 02, 2020).

[43] Nordea. "Et raskt blikk på Nordea." Nordea.no. https://www.nordea.com/no/om-nordea/hvem-er-vi/Et-raskt-blikk-paa-Nordea/?fbclid=IwAR3zlHqoH_Y-3XlJMV5eO1_brjX4nG8vH2f3rBoaUhuXovVWuFKQ-yQ4JTU (Accessed Mar. 02, 2020).

[44] Nordea. "Nåværende rating." Nordea.no. https://www.nordea.com/no/investor-relations/gjeld-og-rating/Navaerende-rating/ (Accessed Mar. 02, 2020).

[45] Nordea. "Basistjenesten." Nordea.no. https://www.nordea.no/privat/vare-produkter/nettbank-og-mobilbank/basistjenesten.html (Accessed Mar. 02, 2020).

[46] Sparebanken Vest. “Bankens historikk.”spv.no. https://www.spv.no/om-oss/om-banken/historikk (Accessed Apr. 21, 2020).

[47] S. Chiasson, C. Deschamps, E. Stobert, M. Hlywa, B. F. Machado, A. Forget, N. Wright, G. Chan and R. Biddle, "The MVP Web-based Authentication Framework," Carleton University, Ottawa, Canada, 2012. [Online]. Available: https://fc12.ifca.ai/pre-proceedings/paper_72.pdf?fbclid=IwAR0d-d4hW9tmj3ysYERI_HA-yyHAXkvkm-BMlfBmUQqPpgaFcepveVnbhgM

[48] J. Biskup, *Security in Computing Systems*, Berlin, Heidelberg: Springer, 2009.

[49] J. J. Stapleton, *Security without Obscurity: A Guide to Confidentiality, Authentication, and Integrity*, Boca Raton, CRC Press, 2014.

[50] Symantec, "Internet Security Threat Report," Symantec Corporation World Headquarters, California, (Volume 24), 2016. Accessed: Feb. 02, 2020. [Online]. Available: https://docs.broadcom.com/doc/istr-24-2019-en?fbclid=IwAR3PsAP1xl8chGCMeb6Dd6Cx-gLLr4pvp7ouHq6jJSea-vOwYtCdA3be1k4

[51] Y. Espelid, L. Netland, A. N. Klingsheim and K. J. Hole, "A Proof of Concept Attack against Norwegian Internet Banking Systems," NoWires Research Group Department of Informatics, University of Bergen, Bergen, 2008. Accessed: Okt. 23, 2019. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.4338&rep=rep1&type=pdf&fbclid=IwAR23oO9FOJ6gt6HfqqmEG9EtSx1BuwdGpAV_eYNapfRe8T3x6s9jEh98USA

[52] K. W. Ross and J. F. Kurose, *Computer Networking: A Top-Down Approach*, 6th ed. USA: Pearson Education Limited, 2012.

[53] M. N. Schmitt, *Tallinn Manual 2.0 on the International Law Applicable to Cyber Operations*, 2nd ed. UK: Cambridge University Press, 2017.

[54] R. J. Anderson, *Security Engineering. A guide to building dependable distributed systems*, 2nded. Cambridge: Wiley, 2008.