



DEGREE PROJECT IN INFORMATION AND COMMUNICATION TECHNOLOGY, FIRST CYCLE
STOCKHOLM, SWEDEN 2016

Modernizing forms at KTH

Using Digital Signatures

PONTUS ENGSTRÖM

Modernizing forms at KTH

Using Digital Signatures

Pontus Engström

2016-10-11

Bachelor's Thesis

Examiner

Gerald Q. Maguire Jr.

Academic adviser

Anders Västberg

Abstract

Today both government agencies and companies struggle to keep up with the pace of the continuous change of technology. With all new technology there are benefits, but new problems might also occur. Implementing new technology for certain tasks may increase both efficiency and security, resulting in a more sustainable work environment. One technology that is increasingly adopted is digital signatures. Instead of using classical handwritten signatures on documents, a digital signature can be more time efficient and have higher security.

In order to implement a digital signature technology some security aspects must be addressed and certain properties ensured. In the document signature process, each time an individual verifies a signature attached onto a document a log entry is created. This log contains information about who verified which document, does it have multiple parts that have been signed, does it need multiple signatures in order to be valid, and at what time and date was the document signed. Logs help to ensure the validity of the document and thereby increase the security provided by the digital signatures.

At KTH, a student must sign an application form with a regular ink-written signature to start a thesis project. This process can in most cases delay the start up to two weeks. This study aims to implement digital signatures for one specific form, an application form for a thesis project. The hypothesis at the start of the project was that the use of digital signature would decrease the time of waiting significantly. Personnel at KTH using digital signature would facilitate their work efficiency, due to less printing and archiving of papers as well fewer meetings.

This study will provide the reader with the necessary fundamental knowledge of cryptography and how digital signatures use this underlying technology. The methodology used in this study was to identify and modify certain software settings, as well collect data from students and personnel at KTH. The collected data was based on time measurements of digital signature processes from students and a faculty member. The results show digital signatures are faster than the current signing process with traditional ink-written signatures. Additionally, the use of digital signatures is expected to reduce the need for printing, transport, and sorting of paper documents. The resulting reduction in use of physical paper should provide environmental benefits.

Keywords: Authentication, cryptography, digital signature, digital certificate, X.509, Adobe Acrobat, LDAP, application form

Sammanfattning

Dagens myndigheter och företag har det svårt att ständigt följa den tekniska utvecklingen. Ny teknik skapar oftast nya fördelar och andra förmåner men kan ibland också orsaka problem. Att implementera ny teknik för specifika ändamål kan öka både effektivitet och säkerhet, vilket resulterar i en mer effektiv arbetsplats. En teknik som introduceras allt mer på sistone är digitala signaturer. Istället för att signera dokument med en handskriven signatur kan en digital signatur vara mer tidseffektiv och ha en högre säkerhet.

För att implementera tekniken bakom digitala signaturer måste särskilda säkerhetsaspekter adresseras och specifika inställningar säkerställas. I signaturprocessen måste varje individ verifiera signaturen som är bifogad på dokumentet, denna verifiering skapar även en logg. En logg innehåller bland annat information om vem som verifierade dokumentet, om dokumentet har fler än en bifogad signatur, behöver dokumentet fler signaturer för att vara giltigt och vilken tid och datum var dokumentet signerat. En logg säkerställer validiteten av dokumentet och ökar därmed säkerheten för digitala signaturer.

På KTH krävs en skriftlig ansökan för att påbörja ett examensarbete. Med nuvarande process kan det i vissa fall leda till en försenad projektstart med upp till två veckor. Den här studien syftar till att implementera digitala signaturer för ett specifikt formulär, en ansökningsblankett för att påbörja ett examensarbete. Hypotesen vid projektstart var att användning av digitala signaturer skulle kunna förminska väntetiden signifikant. Anställda på KTH som utnyttjar digitala signaturer skulle kunna förbättra deras arbetseffektivitet på grund av färre pappersutskrifter, mindre pappersarkivering och färre möten.

Den här studien kommer att förse läsaren med de mest nödvändiga kunskaperna av kryptografi och hur digitala signaturer använder krypteringsfenomenet. Metodiken som användes syftade till att identifiera och modifiera specifika mjukvaruinställningar samt samla in data från studenter och personal på KTH. Den insamlade datan baserades på tidsmätningar av digitala signaturprocesser från studenter, studievägledare och handledare. Resultatet från studien visade att digitala signaturer skulle ge en snabbare signeringsprocess än nuvarande formulär. Det kan dessutom förväntas att med digitala signaturer skulle pappersutskrifter, papperstransporter och sortering av dessa dokument reduceras. Resultatet av minskad användning av fysiskt papper kommer att generera arbetsfördelar.

Nyckelord: Autentisering, kryptografi, digitala signaturer, digitala certifikat, X.509, Adobe Acrobat, LDAP, ansökningsblankett

Acknowledgments

I would like to thank Gerald Q. Maguire Jr for his advice and guidance. His knowledge has been essential to help me shape and complete my thesis, as well as expand my knowledge of computer science. I would also like to thank all students that took their time and created digital ID's, which was the foundation of my data collection.

Stockholm, October 2016
Pontus Engström

Table of contents

Abstract	i
Sammanfattning	i
Acknowledgments	iii
Table of contents	v
List of Figures	ix
List of Tables	xi
List of acronyms and abbreviations	xiii
1 Introduction	1
1.1 Background	1
1.2 Problem definition	1
1.3 Purpose	1
1.4 Goals	2
1.5 Delimitations	2
1.6 Research Methodology	2
1.7 Structure of the thesis	2
2 Background	3
2.1 Authentication	3
2.1.1 Passwords	3
2.1.2 Tokens	3
2.1.3 Biometrics	3
2.2 Cryptography	4
2.3 Cryptographic systems	4
2.3.1 Secret-Key Cryptography	4
2.3.2 Public-Key Cryptography	5
2.4 RSA algorithm	6
2.4.1 Key generation	6
2.4.2 Encryption	7
2.4.3 Decryption	7
2.5 Hash algorithm	7
2.6 Digital Signature	8
2.6.1 Public key infrastructure	9
2.6.2 Public key infrastructure in Sweden	9
2.6.3 Digital Certificate	10
2.6.4 X.509	12
2.6.5 TLS/SSL	12
2.6.6 Cryptographic Message Syntax	12
2.6.7 CMS Advanced Electronic Signatures	13
2.6.8 PKCS #12	13
2.7 Lightweight Directory Access Protocol	14
2.7.1 Schema	15
2.7.2 Attribute	15
2.7.3 Objectclass	16
2.7.4 KTH's LDAP	16

2.8	Adobe	16
2.8.1	Digital Signature in PDF.....	17
2.8.2	Adobe's utilization of PKI standards.....	18
2.8.3	Digital ID	19
2.8.4	PDF file signing.....	21
2.8.5	Features of PDF Signatures	22
2.8.6	Adobe Sign	23
2.8.7	Signature workflows and document storage	23
2.9	Swedish Law regarding Digital Signatures	23
2.10	eIDAS Regulation	24
2.11	Timestamps	25
2.12	Logs	26
2.13	Related work	27
2.13.1	Comparative study of digital signature usage in developed and developing countries	27
2.13.2	Recommendations when implement PKI's.....	27
2.13.3	PDF of student transcripts	27
2.13.4	University use of digital signatures	27
2.14	Summary	28
3	Methodology	29
3.1	Research Process	29
3.1.1	Phase 1: Thorough information gathering phase	29
3.1.2	Phase 2: Modify existing settings.....	29
3.1.3	Phase 3: Implement Adobe's technique as utilized in Acrobat Reader DC.....	29
3.2	Data Collection	29
3.2.1	Sampling.....	30
3.2.2	Sample Size.....	30
3.2.3	Target Population	30
3.3	Experimental Design and Planned Measurements	30
3.3.1	Test Environment.....	30
3.3.2	Hardware/Software to be used	30
3.4	Assessing Reliability and Validity of the data collected	31
3.4.1	Reliability	31
3.4.2	Validity	31
3.5	Planned Analysis of data	31
3.6	Evaluation framework	32
4	Implementation and Result	33
4.1	Creating digital ID in Acrobat Reader	33
4.2	LDAP	33
4.3	Result from the Collected data	35
5	Analysis	43
5.1	Major results	43
5.2	Reliability Analysis	43
5.3	Validity Analysis	44
5.4	Discussion	44

6	Conclusions and Future work	47
6.1	Conclusions	47
6.2	Limitations	47
6.3	Future work	48
6.4	Reflections	48
	References	49
	Appendix A: Application form	53
	Appendix B: Sample form	55
	Appendix C: readme.txt	57

List of Figures

Figure 2-1:	Encryption and decryption of a message.....	4
Figure 2-2:	Both sender and recipient share the same secret key and use it to encrypt and decrypt messages	5
Figure 2-3:	The sender encrypts the message with the recipient's public key, then the recipient decrypts it with the corresponding private key	5
Figure 2-4:	The hash function converts the variable-length plaintext into a fixed-length ciphertext.....	7
Figure 2-5:	How a digital signature is generated and verified using a hash function and the sender's key pair.	9
Figure 2-6:	Above: example of a digital certificate. Below: connection between client and server through SSL.	11
Figure 2-7:	LDAP directory tree using domain-based naming.....	15
Figure 2-8	A digital ID in a signed PDF document (Adapted from "Acrobat DigitalSignatures in PDF" figure 3, page 4 [34])	20
Figure 2-9:	Example of what a Digital Signature might look like in Adobe Reader DC.	21
Figure 4-1:	Example of a student in the LDAP data base	34
Figure 4-2:	Example of a faculty member in the LDAP data base	34
Figure 4-3	Measured time (in minutes: seconds) with 15 different students to create a new digital ID via Acrobat.....	36
Figure 4-4	Measured time (in minutes: seconds) with 15 different students to read and sign a document.....	36
Figure 4-5	Measured time (in minutes: seconds) with 15 different students to create a ID and to sign a document	37
Figure 4-6	Measured time (in minutes: seconds) to create ID, sign document, and to email the next part to sign the application form.....	38
Figure 5-1	The normal distribution of 15 students creation of digital ID displayed as a bell curve.....	39
Figure 5-2	The probability that $200 < X < 300$ is equal to the grey area under the curve (black line is mean value)	41

List of Tables

Table 2:1	Entries in a signature dictionary	17
Table 5:1:	This table displays the mean value, the standard deviation, and the 95% confidence interval of the data collected from 15 students.....	42

List of acronyms and abbreviations

AdES	Advanced Electronic Signature
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
CA	Certificate Authority
CAdES	CMS Advanced Electronic Signatures
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
DAP	Directory Access Protocol
DER	Distinguished Encoding Rules for ASN.1
DIT	Directory Information Tree
DNS	Domain Name System
DSA	Digital Signature Algorithm
EE	End Entity Certificate
eIDAS	Electronic Identification and Signature
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICAs	Intermediate Certificates
IETF	Internet Engineering Task Force
LDAP	Lightweight Directory Access Protocol
MD5	Message Digest 5
OID	Object Identifier
PAdES	PDF Advanced Electronic Signature
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PDF	Portable Document Format
PIN	Personal Identification Number
PKC	Public Key Cryptography
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PPK	Public/Private Key
QES	Qualified Electronic Signature
RA	Registration Authority
RDN	Relative Distinguished Name
RFC	Request For Comment
RSA	Rivest, Shamir, and Adleman algorithm
SKC	Secret Key Cryptography
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
S/MIMIE	Secure/Multipurpose Internet Mail Extensions
TLS	Transport Layer Security
TSA	Time Stamping Authority
TSP	Time-Stamp Protocol
TTP	Trusted Third Party
XAdES	XML Advanced Electronic Signature

1 Introduction

This chapter introduces the problem and its context and then briefly describes the necessary background knowledge. It presents the purpose and goals of this thesis project and what methods were utilized. The chapter ends with a description of the delimitations of this thesis project and the structure of the rest of this thesis.

1.1 Background

The use of an individual's personal signature is still fairly common, but is slowly fading away. Today, most purchases utilize a smart card with an embedded chip. This plastic card and its embedded chip are connected to an account, such as a bank or credit account. To utilize this card to make a purchase an individual must possess the token (in this case the smart card) and know the personal identification number (PIN) for this account. The smart card was invented to decrease payment fraud and identity theft [1] in comparison to the older magnetic stripe card or signature method where an individual only had to give a signature or know some personal information about the account holder [2].

Instead of the classical handwritten signature, other techniques (such as digital signatures) have replaced ink signatures in many areas. Digital signatures have spread worldwide and are widely utilized in e-commerce to prevent fraud and identity theft [3]. Today digital signatures are widely used by governments and internet banking to make services more user-friendly and reliable. However, ink signatures continue to be used for purchases with smart cards and are still a common method for indicating agreement with a printed document.

1.2 Problem definition

KTH is a popular university in Sweden and is one of the leading technical universities in Europe. As with many other universities, KTH still utilizes many forms. These are primarily used when an individual need to sign the form to indicate that they agree to something.

The main disadvantage of the continuing utilization of ink signature on such forms is the large amount of time (in aggregate) spent dealing with such paper forms. Although many people think that the current forms functional well, they are highly inefficient. As a result, a student may need to wait several days to be granted access to a room, permission to start their thesis project, etc. At the same time, there is also a question of security, just how secure are these handwritten signatures and how securely are these paper forms handled and archived. Ink signatures can be forged and misused, which are harder to achieve with digital signatures [4].

1.3 Purpose

This thesis project will examine how to realize digital signatures on forms within KTH. The focus will be to decrease the overall time spent processing these forms and as a result decrease the delay experienced by students and simultaneously increase administrative efficiency at KTH. The last aspect is particularly important because the current processing of forms takes a lot of administrative time. Decreasing the time spent processing these forms would eliminate much of the time that students who have submitted a form spend waiting for the appropriate action(s) to take place. Moreover, implementing digital forms would save KTH both time and money.

This thesis will briefly describe the basics of cryptography, how it is used and why it is needed. It will also depict how important cryptography is for authentication and authorization, i.e., to prove your identity and to securely gain access to some resource.

1.4 Goals

The goals of this thesis project are to create a prototype and suggest guidelines for the KTH administration to implement secure processing of digitally signed forms. The prototype should examine various features such as logs, multi-signing, expiration date of certificates and signatures, etc. These features are essential elements of a secure system for using digital certificates in conjunction with digital signing of forms.

1.5 Delimitations

Note that the choice of any specific commercial software solution made in the context of this thesis project and the selection of this tool for the purposes of this thesis project do not imply any endorsement or expectation that KTH would deploy this specific commercial solution.

1.6 Research Methodology

This thesis project began with a literature study. The background information was mostly found in the book by Jalal Feghhi, Jalil Feghhi, and Peter Williams, *Digital certificates: applied Internet security* [5]. The information in this book was complemented with scientific articles, earlier theses, and other books.

To find a suitable solution I began by examining existing implementations, such as digital signing of documents as realized in Adobe's Acrobat [6]. Adobe has a feature that utilizes public-key cryptography to realize a digital certificate [7]. Data was collected by time measurements from students and personnel at KTH, these results were analyzed in Chapter 5. Other methods will be discussed in Chapter 3.

1.7 Structure of the thesis

Chapter 2 presents relevant information about digital signatures and the cryptography that digital signatures are built upon. Chapter 3 presents the methods used to solve the problem. Chapter 4 presents the implementation of Adobe's technique and the collected data. Chapter 5 presents the analysis. Chapter 6 presents some conclusions and suggests future work.

2 Background

This chapter provides the reader with the fundamental knowledge necessary to fully understand the remainder of this thesis. This knowledge is necessary to understand the methods and how the problem was solved (as presented later in Chapters 3 and 4). The core theoretical background concerns: authentication, cryptography, RSA algorithm, digital signatures, and digital certificates.

2.1 Authentication

An essential part of this thesis concerns authentication. Why do we need authentication, how do we use it and when is it necessary? The following subsections will briefly describe authentication.

2.1.1 Passwords

The most widely used mechanism in authentication schemes is passwords. This method is popular due its efficiency. However, the use of passwords is a source of major vulnerabilities in authentication systems. The greatest issue is that people tend to choose passwords that are short and easy to remember; unfortunately these passwords are often easy for an intruder to guess.

In recent years some users have begun to utilize password managers* (often implemented in a browser or separate software) to store strong passwords. This splits the problem into three parts: secure access to the password manager, the security of the password manager, and the security of the passwords stored by the password manager. The advantage of this approach is that if the user uses computer generated strong passwords and a suitable secure password manager, then the user only needs to know how to securely access their password manager when providing a password. This can significantly increase the security of password based authentication.

2.1.2 Tokens

A token is a physical object that an authorized user possesses. It can be a physical key, an employee badge, smart card, or other similar “things”. Tokens are usually combined with a password, in order to provide higher security and ensure that the token holder is authorized to utilize the token. This combination of methods is called a *two-factor authentication system*, as the claimant must possess both the token **and** the knowledge of the password in order to authenticate himself/herself.

2.1.3 Biometrics

Biometrics is based upon measurements of physiological, morphological, and/or behavioral characteristics of a human to authenticate an individual. The usage of biometrics is commonly used together with other authentication methods, in order to provide higher security and to ensure the individual's identity. Examples of biometric devices are:

- Retina pattern,
- Fingerprint,
- Handprint,
- Voice pattern,
- Keystroke pattern, and
- Signature.

* See for example: <https://1password.com/>

Biometrics are utilized by a number of U.S. federal agencies [8], to support homeland and national security. Further information about biometrics and the use of behavior based biometrics in a single sign-on systems can be found in the recent Master's thesis by BaranTopal [9].

2.2 Cryptography

Cryptography is utilized to ensure safe transfer of information. An early use was the famous Caesar cipher [5], a substitution cipher where each letter in the plaintext is “shifted” by a certain number amount within the alphabet. For example, with a shift of two, A becomes C, P becomes R, and so on. In this case the cycle will “wrap around” after Z, hence Z becomes B. If the plaintext is “BOX”, the ciphertext would be “DQZ”. This cryptographic algorithm does not provide high security, but is the earliest documented form of cryptography. The relationship between plaintext and ciphertext is shown in Figure 2-1.



Figure 2-1: Encryption and decryption of a message

Encryption and decryption of messages are based on utilizing keys. A key is used when a message is encrypted or decrypted and depending on the cryptographic algorithm the keys differ. The two most common types of cryptographic algorithms are *Secret-key algorithm* and *Public-key algorithm*. A *Public-key algorithm* uses a pair of keys: public key and private key. When encrypting text, the public key is used as only a party who knows the private key can decrypt the ciphertext. In contrast, a *Secret-key algorithm* uses one secret key to both encrypt and decrypt the message.

A key together with the algorithm determines how a plaintext message is encrypted into ciphertext, also how to decrypt ciphertext back into the original message. The key is a mathematical value with a certain *key length*. The *key length* is the number of bits or bytes in the key. For example, a 2-bit key has four values: {'00', '01', '10', and '11'}. The number of possible keys is known as the *key space*. Formally the key space is the collection of all possible mathematical values whose representation is less than or equal to the key length. For additional information about the concept of a key space and the effect of the size of this key space on security see [10].

2.3 Cryptographic systems

Two types of cryptographic systems are widely used. These systems have different performance in terms of efficiency, security, and throughput. Both systems will be explained along with their typical uses.

2.3.1 Secret-Key Cryptography

Secret-Key Cryptography (SKC) uses a shared secret key to encrypt and decrypt of messages as shown in Figure 2-2. Provided that this secret key is shared only between sender and recipient, then this scheme provides confidentiality. To exchange this key a variety of techniques may be applied. One of these methods is a Diffie-Hellman key exchange. Secret-Key cryptography is also referred to as *symmetric cryptography*, as both sender and recipient can send messages in both directions. This algorithm is relatively efficient and is therefore most commonly used for bulk encryption of data [5].

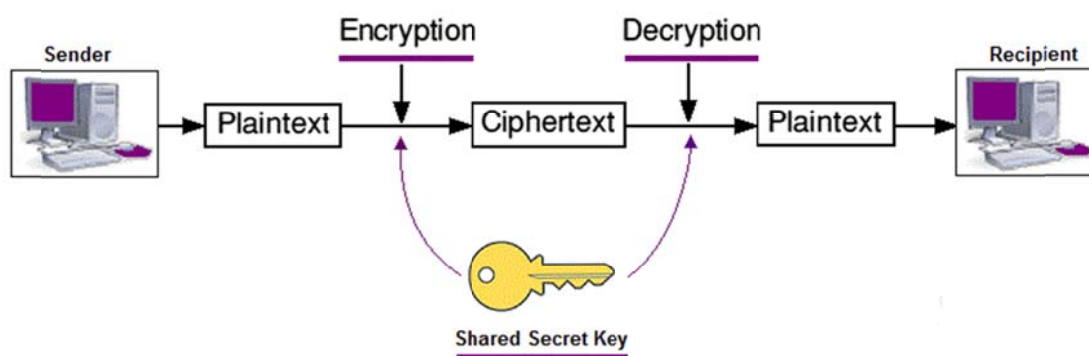


Figure 2-2: Both sender and recipient share the same secret key and use it to encrypt and decrypt messages

2.3.2 Public-Key Cryptography

Public-Key Cryptography (PKC) was invented by Whitfield Diffie and Martin E. Hellman and presented in their 1976 paper *New Directions of cryptography* [11]. PKC is a type of *asymmetric cryptography*, due to using a different key for each direction of message transmission. This type of cryptography is shown in Figure 2-3. As a result, the recipient does not encrypt messages to be sent with the same key as used to decrypt received messages.

PKC uses a *key pair* for encryption and decryption of messages. This *key pair* consists of a private key and a public key. The public key can be made accessible to the public, i.e., anyone can know it, but the private key is kept secret. For example, if Bob wishes to securely transmit a message to Alice and knows that only Alice can decrypt this message, then Bob encrypts the message into ciphertext using Alice's public key. In theory, even if this ciphertext were made publicly accessible only Alice can successfully decrypted the message using her corresponding private key. Hence the message's confidentiality depends upon only Alice having Alice's private key. Encryption and decryption using PKC is about 100 to 1000 times slower than with a Secret-Key algorithm, therefore PKC is rarely used to encrypt large amounts of data [5].

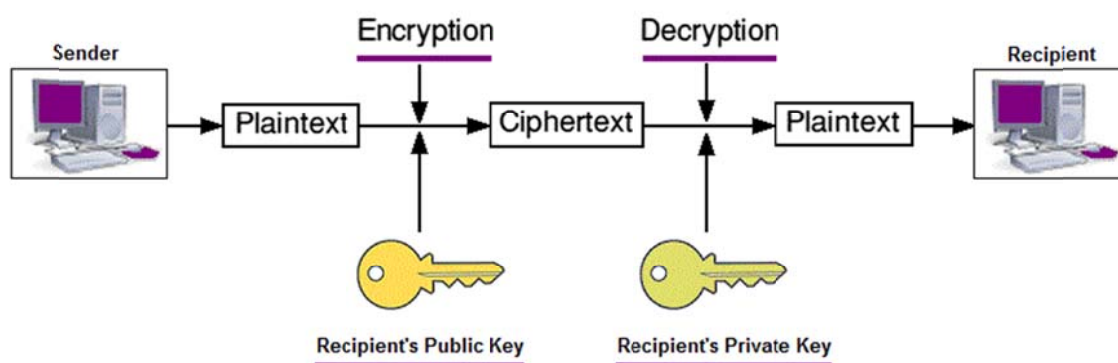


Figure 2-3: The sender encrypts the message with the recipient's public key, then the recipient decrypts it with the corresponding private key

While it is computationally easy to calculate the product of two large prime numbers, it is (believed to be) hard to factorize a large number into its prime factors; hence Diffie and Hellman's Public-Key cryptography and RSA (described in the next section) depend upon the difficulty of integer

factorization (prime factorization) for their security. Consider the case of the prime numbers 149 and 281; their product is relatively easy to calculate: 41869. To determine the prime factors of the number 41869 is hard. However, should a method of rapid factorization become feasible, then both approaches will lose their cryptographic strength.

2.4 RSA algorithm

Rivest, Shamir, and Adleman introduced a Public-Key algorithm named RSA after its creator's [12]. The RSA algorithm generates keys to securely transmit messages over a public network. RSA can be used in three different ways: (1) to provide confidentiality, (2) in digital signatures, and (3) for key exchange. RSA is based on large number factorization (described beneath) whereas Digital Signature Algorithm (DSA) is based on discrete logarithms. These two algorithms have various advantages and disadvantages and they both are utilized in digital signature. However, this thesis will not examine the DSA algorithm.

The process of generating keys will create various *key lengths*, where the key usually are between 256 and 2048 bits [5]. A longer key leads to higher security due to the increased computation needed to perform an exhaustive search of the key space, hence discovering the key by brute force. However, higher security costs in terms of increased computation time and the use of a longer key produces more ciphertext.

The RSA algorithm has three stages: Key generation, Encryption, and Decryption. Each of these will be described below.

2.4.1 Key generation

The RSA algorithm uses two separate keys: a private key and a public key. The public key is assumed to be accessible to everyone and it is used for encryption. To decrypt the message, the corresponding private key is needed. The process of generating a public and private key pair is as follows:

1. Randomly choose two large prime numbers p and q . These numbers should have similar lengths (when encoded as bits) and be kept secret. The prime numbers cannot be equal to each other.
2. Calculate the product of p and q , the result will be stored in n (the modulus for the keys)

$$n = p \cdot q$$

3. Calculate the Euler's totient function:

$$\Phi(n) = (p - 1)(q - 1)$$

4. Choose an integer e , that satisfies:

$$1 < e < \Phi(n) \text{ and } \gcd(\Phi(n), e) = 1$$

where $\Phi(n)$ and e are coprime and do not share factors other than 1

5. Use an extended Euclidian algorithm to find d , such that $d \cdot e \equiv 1 \pmod{\Phi(n)}$

To generate the key pair, the modulus n , exponent e , and exponent d are utilized. The public key is generated using n and e (public exponent). This public key is used for encryption. The private key is generated using n and d (private exponent). This corresponding private key is used for decryption.

In practice the modulus n determines the key length and must be at least 1024 bits to maintain high security (today governments and companies uses 2048 bits key lengths) [5].

2.4.2 Encryption

Once each party had generated a public–private key pair, then the two parties can start to securely communicate with each other. The sender uses the recipient's public key (n, e) to encrypt the message m (potentially with added padding) into ciphertext c :

$$c = m^e \pmod{n}$$

2.4.3 Decryption

The recipient decrypts the received ciphertext c using its own private key to produce the plaintext message m (with padding):

$$m = c^d \pmod{n}$$

2.5 Hash algorithm

A hash algorithm takes input messages with variable-lengths and creates a fixed-length digest as output (as shown in Figure 2-4). The output is called the message digest or a hash. The hash is normally displayed in digits and letters, which makes it impossible for an intruder to read the message or document. If the message has been modified since it was sent from the originator, the hash value will generate to something completely different compared to its previous value. This enlarges the intruder's problems. Another reason for utilizing hash algorithms is for the time efficiency. If the plaintext is quite large a hash algorithm will reduce its length and therefore also reduce the time generating keys (see Figure 2-5). Hash algorithms can also be referred to as message-digest algorithm or one-way hash algorithm. Such an algorithm must satisfy three main properties:

1. It must be infeasible to determine the input message based on its hash, thus the hash function is a one-way function and cannot be easily reversed.
2. It must be infeasible to find an arbitrary message that has a particular hash.
3. It should be computationally infeasible to find two separate messages with same hash value.

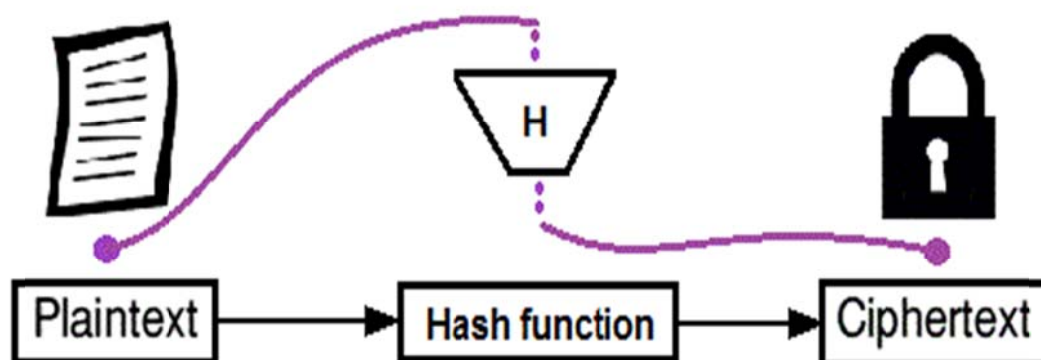


Figure 2-4: The hash function converts the variable-length plaintext into a fixed-length ciphertext

The hash function is an essential function in a digital signature (as will be described in the next section). Example of commonly used hash algorithms are: MD5, SHA-1, SHA-2, and SHA-3. Other common uses of hash functions are to spread items in databases and for securely storing passwords. In the latter example, a hash function is used together with the salt technique to obstruct an intruder

from converting hashed passwords into their original plaintext. This thesis will not examine the details of hashing or salting.

2.6 Digital Signature

A digital signature is an electronic signature or e-signature used to authenticate messages or documents and to ensure that the transmitted data has not been altered. Digital signature is considered to be more useful than e-signature in the area of government, bank or business, because of higher security. An e-signature can be any type of electronic approval method, e.g. electronic sound, symbol, or process. It could be when a program is installed on a computer where terms and conditions agreements are requirement to install the software. A regular user has to accept these terms and thereby pressing the accept button, this is the users signature of approval. These e-signature does not ensure authentication, integrity or security therefore the legal value are not significant high compare to the legal aspects of digital signature.

Depending upon the local laws (see Section 2.9), a digital signature may be considered equivalent to a physical ink signature on paper. However, a digital signature is more time efficient and may offer greater security. A digital signature guarantees the origin and the integrity of a message. More formally, a digital signature ensures:

- Non-repudiation** due the privacy of private key (signing key), the sender cannot repudiate sending the message – as no one else could have signed it
- Confidentiality** due its hash and encryption, any intruder will not successfully decrypt and read a message without the corresponding private key
- Integrity** due to the message's digital signature, the message cannot be altered without the recipient's knowledge
- Authentication** through verification of the certificate, a recipient can verify that a message has been sent by the originator

Figure 2-5 illustrates how a sender generates a signature, then sends both message and signature, and how a recipient verifies the message and signature by comparing the two hashes. In this figure we can see that after computing the hash, the sender encrypts this hash with its private key, thus anyone can verify that the hash of the message as decrypted and the hash of the message as transmitted match, thus ensuring that the message has not been tampered with and that the sender is who it claims to be. This process is called *signature verification*.

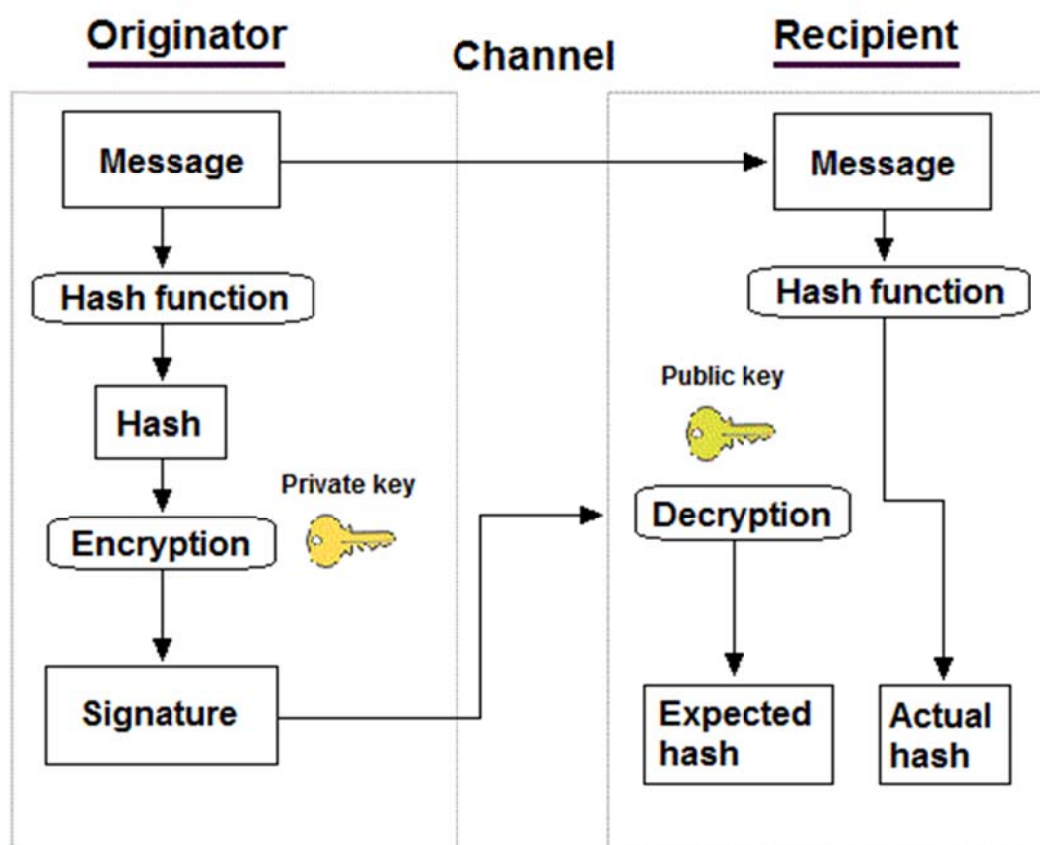


Figure 2-5: How a digital signature is generated and verified using a hash function and the sender's key pair.

2.6.1 Public key infrastructure

In the previous sections we assumed that the recipient can learn the public key of a sender (to verify a signature) and that the sender can learn the public key of the party that it desires to securely send a message to (i.e., in order to encrypt the message before transmission). However, we have not described how the two parties can learn each other's public keys. A public key infrastructure (PKI) provides a framework for authentication of messages or data items, to verify message signatures, and to encrypt messages to be securely sent. This infrastructure has a set of policies, realizes a set of roles, and follows a set of standards to manage the creation, distribution, administration, and revocation of keys and digital certificates. A digital certificate is an essential part of this infrastructure. A digital certificate is used to affirm the identity of a subject and to bind that identity to a public key contained in the certificate. The most commonly uses of PKI are to implement confidential email, e-commerce, and internet banking.

Without a PKI, a sender could encrypt a message to a recipient only if it already has the recipient's public key, but the recipient could not verify the sender's identity. Sensitive exchanges of information rely on the existence and smooth operation of a PKI. Today the popular standard X.509 is utilized as a standardized digital certificate format (see Section 2.6.4).

2.6.2 Public key infrastructure in Sweden

In the current Swedish market two different PKI's are integrated into various systems. Swedish healthcare uses a PKI together with SecMaker's net iD application and tokens such as smart cards to verify employees [13]. This application can be used with single sign-on, thus a user with a valid token and a correct password gains access to a set of applications/databases/services and does not need to

authenticate themselves again (during a single session). This creates an ideal environment for doctors and nurses that on a regular basis sign a dozen journal entries each day. With this technique they do not have to type their password each time they sign a journal (but only enter their password once each time they start a new session or when they re-establish a session). Another benefit is that when an employee removes the token from the smartcard reader the session would normally end, but net iD uses roaming sessions that maintain all unfinished work and simply wait for the employee to continue when the employee re-inserts their card and enters their password. The token is usually put into a smartcard reader built-into the keyboard. When the smartcard is inserted the computer checks to see if this token is valid and known to the system, if so then the user is prompted to enter their password. If the correct password is entered, then the user will be granted access to the system. This session continuity is a particularly useful technique for employees who daily switch between multiple workstations.

The other PKI based solution in the Swedish market is Nexus Personal [14]. This solution is mostly utilized by banks. Nexus Personal is based on several browser plug-in modules that make use of the smart media functions in web applications. This creates an environment for end-users to conduct secure financial transactions, e-commerce, and other security critical services directly from their desktop. If an individual would like to manage documents electronically, he/she must apply for a bankID [15] from their bank. BankID is an electronic identification created to identify a user in a securely way. This technique can be used for managing government documents electronically (tax return, study loan, etc.), log into your bank (without needing an authenticator device or smart card), and digitally signing documents or other agreements. Each time a bankID is utilized it is check against a certificate issued by the issuing entity*. Digital certificates will be described in the next section.

2.6.3 Digital Certificate

A digital certificate is similar to a passport, but consists simply of bits. Given a digital certificate a person, a computer, or an organization can securely exchange information. The use of certificates is facilitated by using a PKI. Each certificate provides identifying information, is forgery resistant, and can be verified by a certificate authority (CA). Another name for a digital certificate is a public key certificate, because the certificate includes identity information, such as the unique public key of the certificate holder. The CA is a trusted entity who issues, manages, and may revoke digital certificates. Examples of CAs include Comodo and Symantec (Verisign) [16]. Almost every CA utilizes a registration authority (RA) who acts as an intermediate in the certificate process. An RA is an authority that verifies user requests and reports to the CA whether the CA should issue a certificate or not. The CA has, in addition to issuing certificates, a responsibility for managing certificates. When a certificate is revoked it is added to a certificate revocation list (CRL). This list plays an essential role in the certificate evaluation process. The list is public and is used to reject what otherwise might be accepted as valid certificates. When a document is signed by a digital signature the recipient checks that the certificate used to sign the document is **not** in this list. If it is in the CRL, then the signature is invalid. The CRL is typically stored in a directory which also contains valid and locked certificates.

Note that if the certificate is not in the CRL, but the date and time of the signature are not within the time period when the certificate is valid, then the signature is also invalid. The question of determining exactly when a document is signed and validating the date and time are addressed in Section 2.11.

Digital certificates are utilized in conjunction with HTTPS (see Section 2.6.5) to enable one-way or mutual authentication as well as secure communication between a client and a server. Figure 2-6 shows an example of a certificate and how it is used. Websites that require high security (e.g. government, bank, and e-commerce websites) will have an Extended Validation SSL Certificate. The URL for sites with such a certificate is displayed by many browsers as a green URL indicating that the

*In addition to the major banks, Försäkringskassan and Centralastudiestödsnämnden (CSN) also participate in this system and can issue certificates.

website is a trusted website and has a valid digital certificate. Right clicking on the green field will give information about the certificate, such as the key length of the public key, whether the certificate uses RSA or DSA, the name of the CA who created and signed the certificate, when the certificate was created, and when the certificate will expire. Depending upon the type of the certificate this information might vary. For example, KTH's web server has a certificate issued by Terena (now GÉANT Association - <http://www.geant.org/>) that has a valid time period from 2015-02-02 to 2017-02-06. This certificate displays that it is using version 3 of the X.509 certificate standard, the key generation algorithm is RSA, and the certificate uses a SHA-256 hash algorithm. These attributes will be described in more detail in Section 2.6.4 where the X.509 certificate is introduced.

By default, a number of certificates of CAs are pre-installed in a computer either during the operating system's installation [17] or built into the web browser. Therefore, a user/computer can know if a website with a given certificate is trusted or not – based upon the implicit trust in those CAs with “built-in” or preconfigured certificates. However, a weakness is that in most cases the end user does not know which CAs they should actually trust.

A certificate can be realized as either a hard or soft certificate depending on where the private keys are located. A hard certificate stores the private key on a smart card, thus these keys are more securely handled than when this key is stored in a file and as a result the certificate is considered to be more trustworthy. In the case of a soft certificate, the private key is stored in a file. This file can be transferred to a USB, diskette, cloud storage, or other type of storage and is generally protected by a password. This type of certificate is normally considered to have lower security than a hard certificate [18]. However, the actual level of security depends upon the encryption used to store the private key.

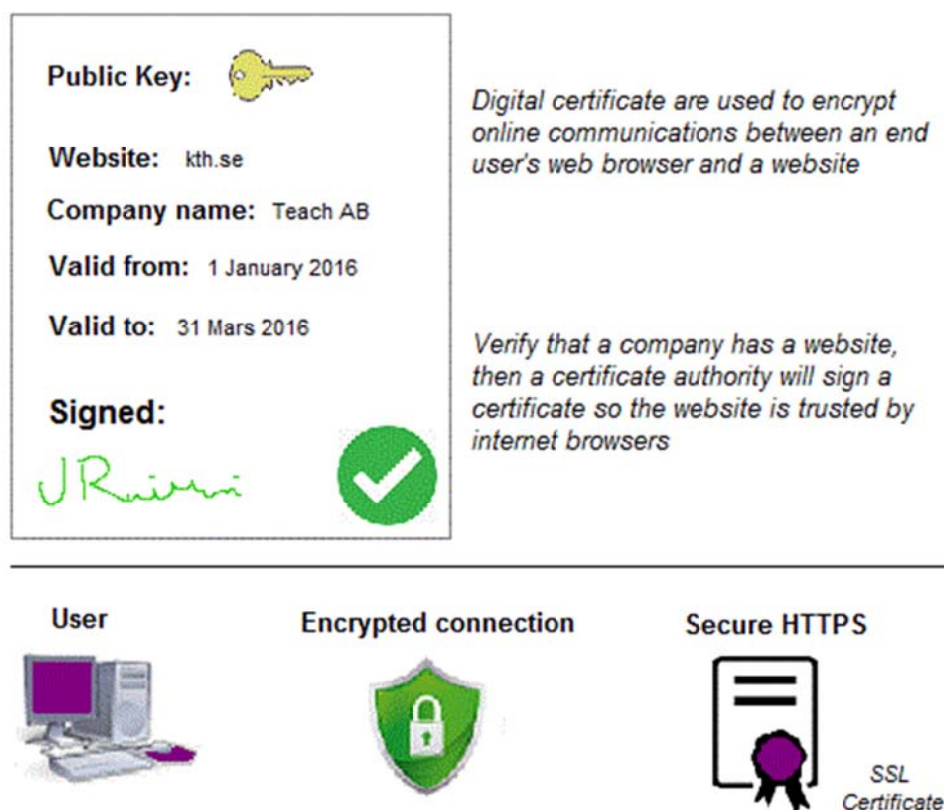


Figure 2-6: Above: example of a digital certificate. Below: connection between client and server through SSL.

2.6.4 X.509

X.509 is a standardized digital certificate format that uses a PKI to verify that a public key belongs to a certain user, computer, or service identity as indicated in the certificate. The X.509 standard is a key means for secure web and email communications. An X.509 certificates includes the following:

Version	indicates which version of X.509 is applicable to this certificate (currently 1, 2, and 3)
Serial number	an unique integer assigned by issuing CA
Signature algorithm identifier	identifies the algorithm (RSA or DSA) to be used to with a signature
Issuer name	identifies which CA has signed and issued the certificate
Validity period	the time interval when the certificate is valid (expressed as a start and end date)
Subject name	the name of the identity the certificate is issued to
Subject public key information	contains the public key material and the identifier of the algorithm
Extensions (optional)	Such as: <i>Issuer unique identifier</i> and <i>Subject unique identifier</i>

An X.509 certificate is used in many forms of cryptography, including: TLS/SSL, Secure/Multipurpose Internet Mail Extensions (S/MIME), HTTPS, and smart cards [19, 20].

2.6.5 TLS/SSL

A Transport Layer Security (TLS) or Secure Sockets Layer (SSL)* certificate is a version of a X.509 certificate, but has extended key usage. An SSL certificate is used together with the SSL cryptographic protocol to provide secure communication over a computer network. For example, as part of HTTPS, an SSL certificate is widely used by electronic commerce web sites enabling users to buy products or services via a web site.

The primary goal of the SSL protocol is to ensure data integrity between two communicating computer applications (hence this is often seen as an application layer protocol). This communication is commonly between a client (a web browser) and a server (a web page) [21]. Note that the strongest form of this security requires that both the client and the server have a certificate that can be validated.

2.6.6 Cryptographic Message Syntax

The Public Key Cryptography Standards (PKCS) are a set of standard public-key cryptography techniques published by RSA Security Inc. in the early 1990s. PKCS #7 (today known as Cryptographic Message Syntax (CMS)) defines a general message syntax that includes cryptographic details, such as digital signatures and encryption. One of the main benefits of CMS is that it allows multiple encapsulations, where one encapsulation (envelope) can be nested inside another. Furthermore, previously encapsulated data can be digitally signed by a certain party. Arbitrary attributes, such as signing time, are allowed to be signed along with the message content. This provides for additional attributes, such as countersignatures to be associated with a signature. Details of CMS can be found in RFC 5652 [22].

CMS supports different architectures for certificate-based key management, where X.509 is the most commonly utilized certificate format. Abstract Syntax Notation One (ASN.1) is a standard which describes rules and structures to represent encoding, transmitting, and decoding data in

*Both are referred to as SSL.

telecommunications. CMS values are generated by utilizing the ASN.1 standard with Basic Encoding Rules (BER-encoding). The values are represented as octet strings (a sequence of bytes).

Another CMS technique is to detach a message's signature. This method is used by S/MIME when sending email. Embedding the signature inside a message has both advantages and disadvantages. An advantage is that embedding the signature in a message requires no support from operating systems or proxy gateways, hence avoiding unintended removal. The main disadvantage is that embedding the signature in a message it will modify the message's semantics.

CMS also forms the foundation for S/MIME which uses encryption and signing to ensure security of authentication, integrity, and non-repudiation of origin. This thesis will not examine the details of S/MIME, the interested reader is referred to [23].

2.6.7 CMS Advanced Electronic Signatures

CMS Advanced Electronic Signatures (CADES) is a set of extensions to the original CMS. CADES extends CMS to provide a general framework for electronic signatures, for use in purchase requisitions, contracts, or invoices[24]. CADES specifies precise profiles of CMS signed data, thus the European eIDAS Regulation (EU 910/2014) is compatible with CADES. The European eIDAS regulation is a regulation for electronic identification and trust services for electronic transactions in the internal EU market. Since July 2014, it is legally binding in all EU member states and if an electronic signature is created in compliance with eIDAS, then this signature has the same legal status as a handwritten signature[25].Section 2.10 gives more about the eIDAS regulation.

If an electronic signature is implemented based on CADES, then it has the status of an *advanced electronic signature*, which has the following requirements:

- it has a unique link to the signatory,
- it has the ability to identify the signatory,
- the signatory is the only one who has control of the data used for signature creation, and
- it can be detected whether the data attached to the signature has been modified after signing

A great property provided by utilizing CADES is that an electronically signed document can remain valid over long periods of time. If the signer or verifying party later tries to deny the validity of the signature, then CADES can be used to rebut this denial.

There exist 3 different eIDAS-compliant implementations of advanced electronic signatures through digital signature: XAdES, PAdES, and (as earlier described) CADES [26]. Each has its own area of application - depending on purpose. This thesis will mostly examine the usage of CADES.

2.6.8 PKCS #12

The Public Key Cryptography Standard (PKCS) #12, also called Personal Information Exchange Syntax, describes a transfer syntax for personal identity information. This syntax can be used for private keys, certificates, various secrets, and extensions. Applications, web browsers, machines, etc., that support this standard create a user friendly environment for users to import, export, and exercise a single set of personal identities. This environment provides benefits to users with different company roles, as they can have multiple digital IDs – each with a different purpose. In Adobe Sign (described in Section 2.8) multiple IDs are an essential tool for many companies. For example, an employee could have administrative roles and also be a part of a project. As another example, a CEO of a national company has great authority within that specific company, but this same person could also be a board member of another international group. Therefore, for many users it is necessary to switch between roles, so that the appropriate ID is used to sign (for different purposes) each type of message

and/or document. This technique of using multiple IDs, each with a specific role can also be utilized even if the user uses different certification methods [7].

As mentioned earlier, PKCS #12 is a standard that uses several privacy and integrity modes to directly transfer personal information. The most secure of these privacy and integrity modes demands that the source and destination platforms have trusted public/private key pairs (to be useable for digital signatures and encryption). When trusted public/private key pairs are unavailable, the standard supports lower-security modes, such as password-based privacy and integrity modes.

PKCS #12 can be implemented in hardware. For example, some hardware implementations offer physical security via tamper-resistant tokens, such as smart cards and Personal Computer Memory Card International Association (PCMCIA) devices [27].

2.7 Lightweight Directory Access Protocol

Lightweight directory access protocol (LDAP) is a lightweight protocol for accessing directory services, specifically X.500-based directory services. LDAP is an Internet Engineering Task Force (IETF) Standard Track protocol that is specified in RFC 4511 [28]. LDAP utilize ASN-1 encoding. Directory queries can be used to access information either in the public "internet" or within a corporate "intranet". LDAP is a "lightweight" version of the Directory Access Protocol (DAP), because the initial version of LDAP did not include security features.

A directory records what and where entities are located along with being able to access attributes of these entities. The domain name system (DNS) is the directory system used to create a relation between domain name and a specific network addressor addresses (and *vice versa*). LDAP allows users to search for information about individuals without knowing in advance where the relevant records are located, while reducing search time. A LDAP directory has a structure similar to a tree hierarchy (see Figure 2-7).

LDAP directories can be distributed over multiple servers. The benefit of this distribution is that each LDAP server* can have a replica of the complete directory by periodically synchronizing with a master copy. LDAP utilizes a *client-server model* where clients connect to servers and make queries. If the server receives a request from a user, it will transfer the request to other LDAP server if necessary and it will ensure a single coordinated response to the user. No matter which LDAP server a client connects to, they all can access the entries in the directory. If a name is presented to one LDAP server, it references the same entry of another LDAP server. This feature is important for a global directory service. LDAP utilizes a special attribute called *objectClass* that specifies which attributes are required and allowed in an entry. The values of the objectClass attribute determine the schema rules the entry must obey. The following subsections describe schemas, attributes, and object classes.

* An LDAP server is sometimes called a Directory System Agent. However, we will not use this term in this thesis to avoid confusion with Digital Signature Algorithm.

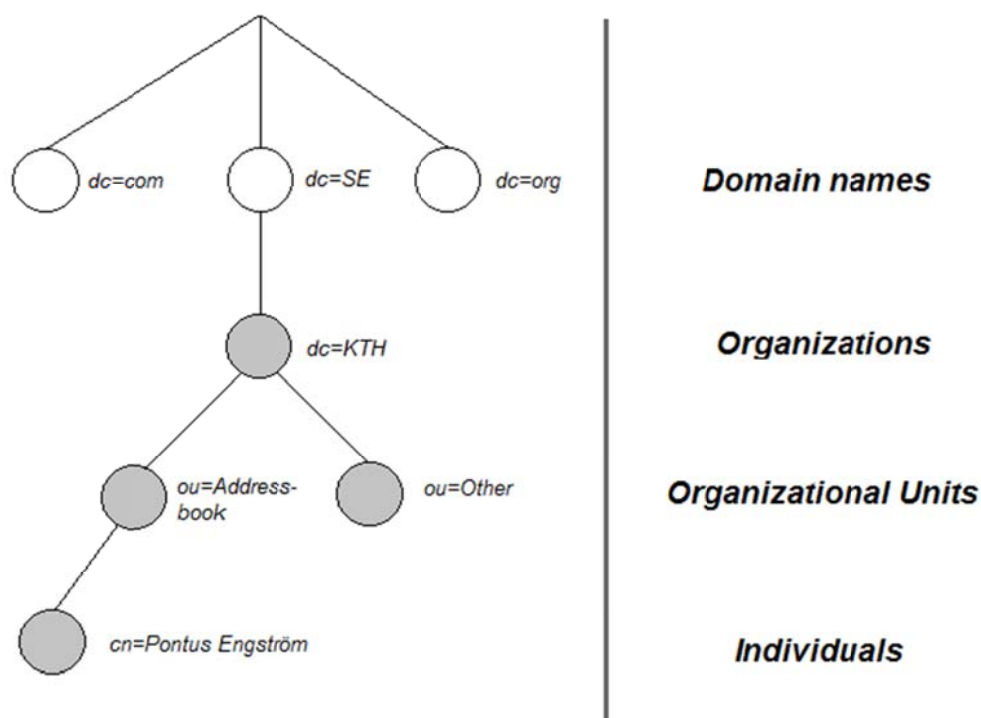


Figure 2-7: LDAP directory tree using domain-based naming

2.7.1 Schema

A schema is a type of document that describes and relates attributes and object classes. To create an object of a certain class, that class must first be defined in a schema. All attributes utilized by an object are required to be defined in a schema. Schemas are written as normal documents, and then converted and inserted into the LDAP database. If the LDAP server cannot find an implementation of the schema, then the object classes and attributes the schema describes will not be utilized [29]. Therefore, it is very important that each element in a schema to be identified by a globally unique Object Identifier (OID). All OIDs uses a hierarchical structure and an organization that uses LDAP or X.500 can create as many branches as they want from their root OID. An OID is a tree structured series of numbers separated with dots (.). An example of an OID is 1.3.6.1.4.1 (iso.org.dod.internet.private.enterprise). This OID is the internet's private enterprise numbering sequence, as assigned by the Internet Assigned Numbers Authority (IANA). Under this branch of the directory, Gerald Maguire at KTH's Telecommunication Systems Lab was assigned branch number 933* and Rickard Schoultz at KTH Royal Institute of Technology was assigned 1664.

2.7.2 Attribute

An attribute usually has a unique name containing some data. Each attribute is a member of one or more object classes. The attributes can have different data types (keyword SYNTAX) such as strings, integer, Boolean, binary, etc. Attributes may be a part of a hierarchy where child attributes inherit all the characteristics of the parent attribute. The hierarchy is used to simplify and shorten the attributes when many attributes share common properties, e.g. maximum length and case sensitivity. Another essential property is that attributes can be optional (keyword MAY) or mandatory (keyword MUST) where the objectclass determines which property it will have. Presenting attributes can either be done

*<http://www.iana.org/assignments/enterprise-numbers>

by single values or multi values. By definition, single means only one data value will be present and multi means that an attribute might appear multiple times in an entry/object class with each instance having a different value. An example of a single value would be the attribute of an email address, where the value can be one or more definitions of the attribute each with a different mail address. Multi values are undesirable for passwords as only one value should be accepted.

2.7.3 Objectclass

Objectclasses are usually containers for attributes where each objectclass has a unique name. As mentioned in Section 2.7.2, the objectclass determines whether the property of the attributes is optional or mandatory. The type of objectclasses can be STRUCTURAL, AUXILLIARY, or ABSTRACT. The keyword STRUCTURAL indicates that a certain objectclass contains attributes and can form an entry in a Directory Information Tree (DIT). A DIT is a LDAP system structured as a hierarchy of objects. Only one STRUCTURAL objectclass in each entry is allowed, but it may be a part of a hierarchy as a SUP, where SUP normally indicates that the objectclass has a parent (superior) objectclass. The ABSTRACT objectclass indicates a non-existent to objectclass utilized for convenience, for example the objectclass *top* which normally terminates an objectclass hierarchy. The last objectclass is AUXILLIARY that include attributes and may be used with any STRUCTURAL objectclass to form an entry[30]. For details of LDAP directory information models see RFC 4512 [31].

2.7.4 KTH's LDAP

A minor part of KTH's LDAP was shown in Figure 2-7, as a simple LDAP directory tree. When searching for entries in this tree some concepts are necessary. The distinguished name (DN) is utilized when an entry is referenced and the Relative Distinguished Name (RDN) is constructed to retrieve the name of the entry itself and concatenate it with the names of its ancestor's entries. In Figure 2-7, the entry for Pontus Engström has an RDN of *cn=Pontus Engstr\C3\B6m* and a DN of *ou=Addressbook, dc=KTH, dc=SE*. To access this information, LDAP defines operations for interrogating and updating the directory. Operations are provided for adding or deleting entries from the directory, modifying existing entries, and changing the names of entries, but most of the time the LDAP is used to search for information in the directory. The search operation allows portions of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria. Additionally, directory services do not provide protection, hence anyone can retrieve information, but usually LDAP provides a mechanism for a client to authenticate, i.e., prove its identity to a directory server. Some common uses of LDAP are machine authentication, user authentication, address books, and email address lookups. LDAP supports integrity and confidentiality services. An example of an address book entry is shown in Section 4.2.

Two critical jobs that LDAP performs are providing an authentication database and once the user has been identified this authentication controls access to resources, applications, and services. Another use for LDAP is to provide a central place to store usernames and passwords. This creates an environment where many applications and services connect to the LDAP server to validate users. This makes it easier to update and change user passwords as there is only one logically centralized place they are stored [32].

2.8 Adobe

Adobe Systems Incorporated is a leading company in information technology. Some of their currently popular applications are: Photoshop, Acrobat Reader DC, and Adobe Flash Player. Quite recently Adobe launched a new feature called Adobe Sign, whose purpose is to enable a user to digitally sign a document or Portable Document Format (PDF) file, to speed up transactions, and to better visualize the entire process when compared to earlier. According to Adobe, utilizing their application will

increase sales, improve human resource experience, keep procurement documents digital, and integrate everything seamlessly into a specific business system.

As an example application, this thesis will use the KTH application form for a degree project (see Appendix A) and utilize the functions embedded in Adobe Reader DC (the implementation is described in Chapter 4).

2.8.1 Digital Signature in PDF

Documents can have many different formats*, but the most popular and useful one in the context of forms is PDF. In a PDF file all of the signature information is stored in a *signature dictionary*. The dictionary contains entries that are required or optional, where the required entries are summarized in Table 2:1. All of these required entries are managed by a *signature handler*. The optional entries can either be used or omitted, but developers are encouraged to manage them in a standard way if they are used. It is suggested that all private entries be prefixed with the registered handler name followed by a period (.), to avoid name duplication.

Signatures are generated by computing a digest (hash) of the data in a specific document, and storing this digest in the document. To verify the signature and to verify that the document has not been tampered with, the digest is recomputed and compared with the one stored in the document. If the calculated digest does not match the one stored in the document, then the document has been altered since it was signed. There exist two techniques for computing a reproducible digest of the contents of all or some parts of a PDF file:

Byte range digest Computed over a range of bytes in the file, indicated by the ByteRange entry (see Table 2:1). Usually this range is the entire file, including the signature dictionary, but excluding the signature value itself (Contents). When a byte range digest is present, the signature dictionary's values have to be direct objects.

Object digest Computed by selectively walking a subtree of objects in memory. Starting with the referenced object which usually is the root object. The resulting digest, with other information about how it was computed, is placed in a signature reference dictionary.

Table 2:1 Entries in a signature dictionary

KEY	TYPE	VALUE
Filter	Name	The name of the preferred signature to use when validating the signature [†]
Contents	Byte string	The signature value
Cert	Array or byte string	An array of byte strings representing the x.509 certificate chain utilized when signing and verifying signatures that use PKC, or a byte string if the chain only has one entry [‡]
ByteRange	Array	An array with pairs of integers describing the actual byte range for the digest calculation

* These different format files typically have different files extensions, such as .docx, .xlsx, .pptx, and .txt.

[†] Examples of signature handlers are Adobe.PPKLite and VeriSign.PPKVS.

[‡] The Cert entry is only required if the optional entry SubFilter is adbe.x509.rsa_sha1 appears.

The signature reference dictionary contains entries indicating how the object digest was computed along with other information. Here the TransformMethod and TransformParams are essential entries. The TransformMethod specifies the method utilized to compute the digest, whereas the TransformParams specifies the variable portion of the computation. More information can be found in the PDF reference: Adobe® Portable Document Format, Version 1.7 [33].

Table 2:1 briefly describes the entries in the signature dictionary when a byte range digest is computed. Additionally, in the Contents entry, the signature value will change depending on whether ByteRange is present or not. If present the value is displayed as a hexadecimal string representing the byte range digest value. If not present, then the value is an object digest of the signature dictionary, excluding the Contents entry. The Contents value is usually either a Distinguished Encoding Rules for ASN.1 (DER) encoded PKCS#7 binary data object or for a public-key signature it is a DER-encoded PKCS#1 binary data object.

In the Cert entry, the first part of the array must be the signing certificates well as other certificates that are used to verify the authenticity of the signing certificate. These certificates can subsequently be used to verify the signature value in Contents. If the optional entry SubFilter is adbe.pkcs7.detached or adbe.pkcs7.sha1, then this entry is not used and the certificate chain must be placed in a PKCS#7 envelope in Contents.

In the ByteRange entry, the array contains the starting byte offset and the length in bytes. Multiple discontinuous byte range are used to describe a digest that does not include the signature itself (Contents entry) [33].

The Acrobat family supports all of PDF's features and at a high level these signature features can be grouped into following categories:

- Add a digital signature to a document,
- Check signature for validity, and
- Control the signature workflow with permissions and restrictions.

2.8.2 Adobe's utilization of PKI standards

A PDF file's digital signature utilizes PKI standards as PKI solutions are widely deployed in both business and government settings. In a specific PDF signature workflow, the PKI is assumed to contain information about the digital ID issuers, users, administrators, and different hardware or software systems used in the workflow. PDF viewers, such as Adobe Reader DC, are designed to provide a user-friendly environment with which to interact with all of these components [34]. The Adobe document "Digital Signatures Workflow Guide: A guide for workflow owners" [35] describes how the overall signature workflow operates, starting with designing documents to be signed and ending with how to customize workflows.

Handwritten signatures usually need a trusted authority, e.g. a notary public, to witness the signing of an important document. The notary is assumed to be trustworthy and hence will not be questioned as an authority, therefore the signature which they have notarized is valid and there exists a chain of trust in the handwritten signature process. A digital signature using a PKI will provide the same sort of chain of trust. In this model the certificate authority acts in a similar fashion to the notary public in notarizing a signature – in this case by the certificate authority issuing (and signing) a certificate.

According to Adobe, the following PKI components are directly related to providing trust [34]:

Certificate Authority (CA)	A CA is an authority that sells or issues digital IDs. The CA signs its own certificate and is usually the root certificate at the top of a certificate chain. The CA is further described in Section 2.6.3.
Intermediate certificates (ICAs)	An authority who acts as an intermediary between the end entity and the root certificate. The ICA provides services such as policies, timestamping, revocations lists, etc.
End entity certificate (EE)	The user's/signer's certificate which is the last element in the certificate chain.
Digital ID	A digital ID is an ITU-T X.509 v.3 representation of the data usually associated with a person or entity. This digital ID is generally stored in secure manner, such as a password protected file on a computer, smart card, USB storage device, etc. This digital ID contains a public key certificate, a private key, and some additional data.
Public key certificate	The public-key certificate includes the public key portion of a public/private key pair together with attributes and associated extensions (for example, indicating the certificate's owner, validity period, and allowed usage).
Private key	Key generation produces a public and private key pair. The secret key is used to validate incoming messages and sign outgoing ones.

These entities are central to any PKI where the PKI also includes other company-owned and third party items. The PKI administrator manages the creation and distribution of digital IDs, LDAP servers, timestamp servers, revocation lists, and other items. The PDF language supports the data needed to interface with these specific components.

Note that the key generation can occur at the user, with the resulting public key signed by the CA, or there may be one trusted party that both generates the key pair and signs the certificate for the public key. This latter approach is very common for the case when the key pair and certificate are stored on a smart card, as this allows the vendor who produces the smart card to fill it with a generated key pair and signed certificate. However, it has the risk that this party knows the private key.

2.8.3 Digital ID

Of the previously explained PKI components there is one component worthy of a more complete description: Digital ID. A digital ID is data associated with a person or entity, similar to an electronic driver's license that proves your identity. It usually contains information such as personal name, email address, certificate authority (CA) that issued it, a serial number, and expiration date (start and end date). The digital ID is made up of two keys: the *public key* - locks or encrypts data and the *private key* - unlocks or decrypts data. When signing a PDF file, the private key is applied to create the digital signature. The public key is stored in a certificate that is distributed to others. Normally when a PDF file is signed the user distributes this certificate to others so that they can validate the user's signature and identity. As mentioned in Section 2.8.2, a digital ID is generally stored in a password-protected file. This is essential because the private key is the only component that can unlock information that was encrypted with the certificate. Figure 2-8 shows (on the left) a digital ID stored in a computer or other device and (on the right) a signed PDF document with the certificate embedded in it.

In Adobe Acrobat Reader DC, a user could either self-sign their digital ID or request a signed digital ID from a CA. Once the user has a digital ID, the user can sign a PDF file.

Depending on how the digital ID will be utilized there are benefits with each of type of certificate. When self-signing a digital ID, the digital ID is generated by the Adobe software itself upon a user's request for a new digital ID. For example, a user can create a digital ID using Adobe Acrobat. The self-signed certificate and an ordinary user's identity are bound to a public key generated by the application. When the certificate is generated it is signed by the user's own private key. This is very time efficient and useful in smaller business where the parties have already established mutual trust. Note that there is nothing that prevents someone from generating a self-signed certificate with someone else's name. As a result, a self-signed certificate does not provide a high level of assurance. When documents require greater trust (which occurs for larger companies, banks, or other markets), then a third party, in this case a certificate authority is introduced to validate the identity of the user and to issue a signed certificate. This CA (e.g. Symantec or Entrust) is trusted by many companies. It is important to choose a CA that is trusted because they have the responsibility for verifying the true identity of a certificate holder to others. Adobe has a list of trusted companies that offer digital IDs at <https://helpx.adobe.com/acrobat/kb/approved-trust-list1.html>.

The CA business is fragmented. However, there are a number of national providers that dominate their home market. This is because many uses of digital certificates are linked to the national laws and regulations. In Sweden, SP Technical Research Institute of Sweden* is a major CA who issues certificates to individuals and companies.

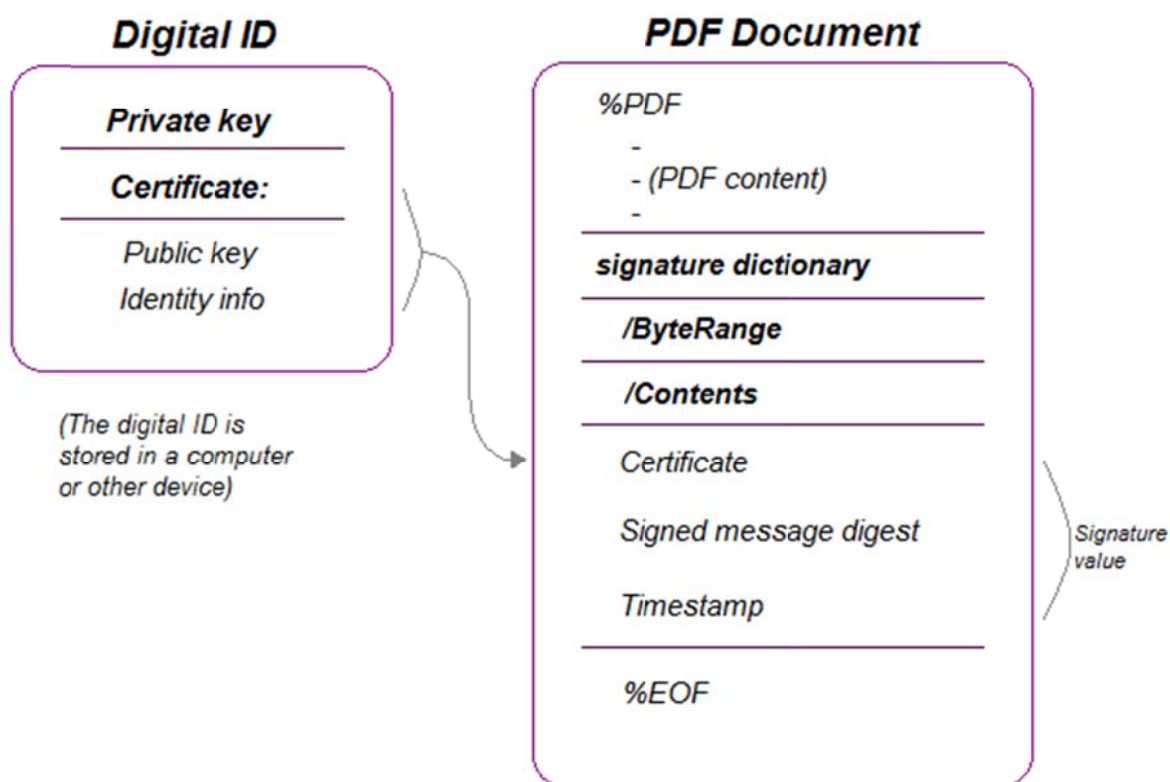


Figure 2-8 A digital ID in a signed PDF document (Adapted from "Acrobat DigitalSignatures in PDF" figure 3, page 4 [34])

* For more information visit: <https://www.sp.se/sv/units/certification/Sidor/default.aspx>.

2.8.4 PDF file signing

As mentioned in Section 2.8.3, signing a PDF file will embed the signature into the document itself. Formats other than PDF might require two different applications to handle the document and the signature; and might even need to manage two separate files for each signed document. In contrast, by embedding the signature into the PDF document a viewing application can render the document for viewing and can even validate the signature. Additionally, some documents will have parts which are signed by different parties; hence the PDF viewer might enable the user to perform some PDF modifications *without* invalidating the signature of a part of the document. This can be very useful in many scenarios – such as a workflow where multiple people read and sign the document.

The relationship between a digital ID stored on the user's device and the signature embedded in the PDF document was shown in Figure 2-8. This figure does not include additional information about the signature's value, such as signature graphic, time stamp, and other data (see Figure 2-9). The presence of this other information depends upon the purpose of the document.



Figure 2-9: Example of what a Digital Signature might look like in Adobe Reader DC.

All signed PDF documents are associated with a signature handler, as mentioned in Section 2.8.1. The signature handler that is built into Adobe Acrobat utilizes Public/Private Key (PPK) cryptography technologies (asymmetric cryptography – as described in Section 2.3.2). The signature dictionary contains the name of the signature handler that will be used to process that specific signature.

The process of signing a PDF file involves the following steps [34]:

1. The document is turned into a stream of bytes to be signed.
2. The PDF file is written to disk with a suitably sized space left for the signature value, according to the worst case values in the ByteRange array. The ByteRange array contains four numbers, e.g. [0, X, X + space for signature, Y]. The first number in each pair represents the offset from the beginning of a stream of bytes that will be included when calculating the hash (the start offset of a stream is byte number 0). In this example the first pair includes [0, X] where X is the stream length, the second pair includes [X + space for signature, Y] where Y is the stream length of this second part of the document. Together these two pairs define the sequences of bytes and what is to be hashed. This means that the hash is calculated for bytes 0 through X-1, and bytes X + space for signature through X+Y*. The actual signature value is stored in the /Contents key between the end of the first part of the file and the second part of the file.
3. Because the byte offset cannot change, therefore we must initially allocate extra bytes. The space for the entire hash value is initially zeroed. After the signature value is calculated, the area specified for this signature by the ByteRange array is overwritten using the correct values.

*This four-tuple is due to backward compatibility when two parts of the file could be included in the hash, but since PDF 2.0 the hash is always computed over the whole file excluding the space allocated to put the results of the hash.

4. The hash over the bytes of the document as specified by the ByteRange array is computed using SHA-256 hash algorithm. Adobe Acrobat always computes the hash of a document signature over the entire PDF file, starts from byte 0 and ending with the last byte in the physical file, but excludes the signature value bytes by treating these bytes as if they contain the value zero.
5. Encrypt the hash value with the signer's private key and then generate a hex-encoded PKCS #7. Signature object.
6. Place the signature object into the PDF file by overwriting the placeholder/Contents value. Space not used for the signature object is overwritten with zeros.
7. Finally, the PDF document is re-loaded into Adobe Acrobat to ensure that both the in-memory and disk-version are identical.

Signature validation is an important part of the signature process. Signature validation is done by verify the signature using the entity's public key from its certificate. When signing a PDF document, the signer's public key is embedded as a part of the signature, therefore the recipient will always have the public key when they have the document. To validate this signature, the signer's certificate is retrieved by the validator and compared to their list of trusted certificates. This validation of signature methods proceeds according to the following steps:

1. The recipient's application generates a one-way hash of the document using the same hash algorithm that the signer used, while excluding the signature portion of the byte stream.
2. The encrypted hash value contained in the document is decrypted using the signer's public key.
3. The decrypted hash value is compared to the locally generated hash value.
4. If there is a match, then the signature is validated and the document is known to be signed by the owner of the certificate. Additionally, the recipient knows that the document has not been modified since the time it was signed.

Note that there are differences between a trusted signature and a valid one. The trust level of the signature on the document depends on the recipient's application configuration; whereas a valid signature derives from the CA's signature of the certificate [34].

2.8.5 Features of PDF Signatures

The PDF language's signature features support several useful techniques when signing a PDF document. One worth mentioning is support for biometric identification, such as handwritten signature, fingerprint, or retinal scan. The signature handler checks the data retrieved for authentication according to the rules defined in the ISO PDF standard [36].

Additionally, it is useful to have several different digital IDs in order to use them for different purposes. Section 2.6.8 described the impact of using different ID in an employee's different roles in the company on the business market or other markets. As mentioned in Section 2.8.3, Digital IDs are usually password protected and can be stored in PKCS #12 file format on a computer, token, or smart cards.

Another feature is a roaming ID account. A roaming ID is a digital ID that is stored on a server and can be accessed from anywhere with an internet connection. This requires that the user have an account with an organization that supplies roaming digital IDs.

In a large organization a directory server is used to access all of the different certificates relevant to the organization. The directory server(s) usually act as centralized repositories of identities within an organization. Such a directory server is an excellent place to store user certificates in those

enterprises that use certificate based encryption. Directory servers enable the user to locate and retrieve certificates from LDAP servers or other network attached servers. When a certificate has been located and utilized, it can also be cached as a trusted identity in order to save time if this certificate is needed in the future. This requires a storage area for trusted certificates. This storage area can even be shared by a workgroup, thus facilitating the use of encryption by different users within the same workgroup. Note that using cached certificates requires another process to receive certificate revocation announcements and purge the revoked certificates from the cache.

Another great feature is the usage of time stamps. If a user signs a document with time and date it reduces the possibilities of invalid signatures. A user can review all the signed documents and check if the time when a document was signed is valid for this user and this signature. This is similar to the use of a log, where users can double check the signature with the corresponding time. Adobe utilizes this technique and users can apply time stamps to their certificate-based signature. If the timestamps are associated with a trusted time stamp authority's certificate, they are easier to verify. Obtaining a timestamp is usually done from the same certificate authority who issued the digital ID, but one can also use a third-party timestamp authority. Timestamps can also be used together with block chaining to produce evidence of the temporal order of documents, see for example [37]. Further details about timestamps are given in Section 2.11.

2.8.6 Adobe Sign

Adobe Sign, formerly Adobe EchoSign, is a technique developed to manage e-signatures and to make it possible to sign documents on different devices in a secure and legal way. The goals of Adobe Sign are to facilitate company's management of documents and to create an environment for faster agreements (both internal and external). The idea is to digitally sign documents where ever you are, sign on different devices, and even switch between multiple users. Adobe Acrobat Reader DC needs to have been pre-installed on the device to utilize the Adobe Sign technique and to be able to digitally sign documents. For an individual in Sweden utilizing Adobe Sign costs 1275 SEK/year. Enterprises must contact Adobe to get a tailor-made version, these have different prices. Adobe Sign is a part of the *Adobe Document Cloud*. This cloud service includes Adobe Sign and other functions, see <https://acrobat.adobe.com/us/en/acrobat.html> for more information.

2.8.7 Signature workflows and document storage

Today most companies and enterprises that are digitally integrated use some kind of cloud service. Some well-known services (such as Salesforce, Workday, SharePoint, and Microsoft Dynamics CRM) are usually already integrated in the company's workflow. To add digital signatures, the company adds Adobe Sign into the mix. The integration steps are basically none, because Adobe Sign is compatible with the most well-known cloud services available on the market.

2.9 Swedish Law regarding Digital Signatures

When digital signatures increased in popularity some rules and guidelines had to be created. These guidelines and rules became laws to the benefit of governments and banks. In 1996, the UN published the UNCITRAL Model Law on Electronic Commerce that created global standards for digital signature legislation for e-commerce and e-business. This established a solid basis for governments and banks to implement this technology. It did not take long until Europe created similar legislation: European Directive 1999/93/EC. This legislation was established in 1999 and was the primary electronic signature law. This law included both clarification and protection for e-business as well for the digitally signed documents. All countries that were members of the European Union were obliged to implement the law.

In Sweden, after the introduction of the European Directives in 1999, the Qualified Electronic Signatures Act (SFS 2000:832) [38] was established, thus making an electronic signature a valid authentication ID just as a handwritten signature - with the exceptions of deeds. Other scenarios when an electronic signature cannot be used are described in [39].

The purpose of the Qualified Electronic Signature Act (SFS 2000:832) is to facilitate the use of electronic signatures. The law will facilitate the use of e-signatures, by providing for secure signature creation devices, qualified certificates for electronic signatures, and the execution of these certificates. The law is mostly directed to Swedish companies who issue certificates to public users. The law provides for two signatures: electronic signature and advanced electronic signature. In the case of an electronic signature, data in electronic form attached to other electronic data is used to verify the content derived from the originator and that this content has not been altered. An advanced electronic signature is an electronic signature associated exclusively with a signatory, making it possible to identify the signatory, because it was created using only information that the signatory controls. Such an advanced signature can be attached to electronic data to ensure that any adulteration of this data can be detected.

In Sweden, the only organizations that issue a certificate are banks, with the exception of Telia who issues certificates for government entities, such as the Tax Authority (Skatteverket). If an individual wants to utilize digital signatures, he or she generally contacts a bank and applies for a bankID*. A bankID is the leading electronic identification in Sweden. Depending on the bank, some requirements may need to meet before the bank will issue a certificate. Some banks require that the customer visits a bank office to sign agreements via paper documents. Other banks, such as Swedbank, only require that the customer logs into the bank's webpage with a unique security token. This token is used to verify the identity of a user based upon the user supplying the correct password, establishing that the user has the authority to access his/her private webpage. This webpage is usually called *internetbanken* where customers can pay their bills, transfer money, or apply for specific applications such as bankID or Swish.

In Sweden there are three methods of using a digital signature: from a file, from a smart card, or on a phone. The utilization of mobile phones with digital signatures has rapidly increased since the launch of mobile ID and even more so lately because of the Swish application. For further details about the Swish mobile to mobile payment system (owned by Danske Bank, Handelsbanken, Länsförsäkringar Bank, Nordea, SEB, and Swedbank) see <https://www.getswish.se/>.

For more detailed information about the current laws for electronic signatures in Sweden, see the documentation from the Swedish government regarding digital signature and its legal aspects [40].

2.10 eIDAS Regulation

As mentioned in Section 2.9, the introduction of the European Directive 1999/93/EC was necessary to enable the utilize of electronic signatures when signing agreements. The negative aspects of this directive was that members of the European Union (EU) used the directive with different guidelines resulting in different legislation. Some countries, such as Sweden and Austria, adopted very strict versions, while the United Kingdom adopted a similar (liberal) version to that used in the USA. The European Directive failed with its intention, that all member states should use the same set of technical standards [41]; therefore, it was essential to change the directive to unite all national members.

In 2011, one of the top priorities in the European Commission was to adjust and renew the current directive. The purpose of this revision was to create a single European digital market that all members of EU would use, to ease the electronic interactions between businesses, citizens, and public authorities. Three years later a review of a new regulation was conducted where its main purpose was to ensure confidence in electronic signatures and to create a mutual recognition of electronic

* For more information about bankID, see <https://www.bankid.com/en/om-bankid/detta-ar-bankid>

signatures across the Europe Nation. This new regulation is called Regulation No 910/2014, and unlike its predecessor it is a regulation rather than a directive. On the 1th of July 2016 the new regulation will repeal the current electronic signature directive and automatically replace the inconsistent national laws in Europe with a single regulation.

The main impact of the new eIDAS regulation is partly how it changed the definition of different signatures. The definition of a regular electronic signature remains intact and unchanged - an electronic signature should not be denied legal effect on the grounds that it is in electronic form, but modifications have been made to the Advanced Electronic Signature (AdES) and the Qualified Electronic Signature (QES).

AdES utilizes CAs to verify whether certificates are valid and that the signer of a document is the person they claim to be. This method has been used for many years, but with the new eIDAS regulation enables the signer to use the latest technologies, such as mobile devices, to perform these functions.

A QES is similar to the AdES, as they both are uniquely linked to the signer. However, QES is based on *Qualified certificates*. A Qualified certificate can *only* be issued by a CA which has been accredited and supervised by specific authorities designed by the EU member states and satisfies the requirements of eIDAS. Another important factor is that Qualified certificates must be stored on qualified signature creation devices such as smart cards, on USB tokens, or by cloud based trust services. A QES is the only type of electronic signature that has the same legal equivalence as a handwritten signature. Another important aspect of QES is that this type of electronic signature will ensure mutual recognition of its validity by all member's states in EU. This is a crucial step to create a united digital market across the entire EU.

The final concept that will be introduced by the new eIDAS regulation is electronic seals. These seals are similar to electronic signatures, but they are only available for legal entities, such as corporate entities [41]. Seals are supposed to be used by an authorized signer for a specific *entity*. These seals are associated with a specific entity and any use of a seal will be presumed to be binding on that entity, especially in case of *qualified electronic seals*. This concept can be considered to be quite interesting; however, how it will be utilized and whether it will be utilized at all are open questions. For more detailed information about the new regulation, see [42].

2.11 Timestamps

An important element of digital signatures is that they can be timestamped, thus it is possible to prove that a document has been signed after some date & time and before some other date & time. According to IETF's Time-Stamp Protocol (TSP), specified in RFC 3161 [43], a trusted timestamp is a timestamp issued by a trusted third party (TTP) who acts as a Time Stamping Authority (TSA), similar to a Certificate Authority (see Section 2.6.3). Timestamps are supposed to strengthen the digital signature procedure as the signing entity cannot backdate a timestamp or repudiate the signature. This increases the reliability and reduces the vulnerability of a digital signature. For even higher security it is possible to utilize multiple TSAs.

Creating a timestamp is based on the technique of digital signatures and hash functions. When a signature has been attached to a document a hash will be calculated. This hash is sent to the TSA where they concatenate a timestamp to the hash and calculate a new hash using the TSA's private key. The concatenated value of a signed hash and a timestamp is sent back to the requester (the person who requested a timestamp). Another notable function when creating a timestamp is that the original data (document) cannot be calculated from the hash (as it is a one-way function), thus the TSA never sees the original data only the hash – thus ensuring data confidentiality. Moreover, it is assumed that it is difficult to generate another document that has the same hash; hence the signed hash can be used to prove the existence of a particular document (and its contents) at a specific point in time.

Once a timestamp has been created it is important that one can check its validity. Trusting the TSA is essential to verify that a document was *not* created *after* the timestamp was created. Additionally, the requester of the timestamp cannot repudiate being in possession of the original data at the time given by the timestamp. The TSA's role as a TTP is similar to trusting CAs (some CAs can serve as a TSA), therefore it is important to use well-known authorities for higher security (e.g. for banks and governments). However, to ensure that a document has not been backdated or to prevent repudiation of the timestamp of the original document, hashes must be calculated and compared. The first hash value, let us call it Hash X, is a concatenated value of the original hash value from the digital signature and the added timestamp from the TSA. The other hash value, Hash Y, is calculated from the signed hash by using the TSA's digital signature (signed by the TSA's private key). Hash Y is decrypted and validated using the TSA's public key. If Hash X and Hash Y match, then the document's timestamp and message are unaltered and the timestamp was issued by the stated TSA. If there is not a match, then either the timestamp was altered or the timestamp was not issued by the TSA.

Some electronic signatures with higher value need to be valid over a longer period of time. This is to guarantee that a signer cannot repudiate signing a document even many years after it was signed. For more detailed information, see [44].

2.12 Logs

Logs can be defined in different aspects depending on the environment. In Adobe, logs are created for users to troubleshoot problems where a workflow or plugin does not work when Protected Mode is enabled (Protected Mode is set in Adobe Reader DC by default and works as a "read-only" mode, it is to block any potentially executable files to be executed or write to system directories without the user's knowledge). The created log may provide guidance to whether use a custom policy file and to re-enable the broken workflow or plug in.

Privileged locations are a feature created to improve the workflow. A user can add specific files, folders, and hosts to privileged locations to trust PDF files, and to be able to bypass the security restrictions[45]. In the Adobe Reader DCs preference dialog box there are several options to choose from depending on what security level the workflow is most suitable for. One worth mention is "Automatically Trust Documents With Valid Certification", with this preference the user trusts that the document is sent from a trusted user. Note that the one exception to such trusted PDF's parity with privileged locations is that this level of trust does not apply when the PDF is viewed in Protected View. The TrustManager handles these applications where staff administrators are advised to review these settings and configure them to what best satisfies their workflow. In the previous example of privileged locations, the attribute "bEnableCertificateBasedTrust" is by default set to zero (or null), but do make certified documents equivalent to privileged locations (trusted) it must be set to one.

Signature Validation Logging is a feature that Adobe Reader versions 8.x and later enable logging certificate validation and revocation checking information. It is to set both logging level and log location. To set the log location the path must already have been created. Additionally, when Protected Mode is enabled the log file path must be permitted by the Protected Mode. An example of chain building log file settings could be similar to:

```
[HKEY_CURRENT_USER\Software\Adobe\Adobe
Acrobat\8.0\Security\cASPKI\cAdobe_ChainBuilder]
"iLogLevel"=dword:00000008
"sLogFilepath"=(BINARYpathtoexistingdirectoryforlogfile)
```

Where the `iLogLevel` specifies the log level during chain building and validation, and the `sLogFilepath` specifies the full path of the text log file, e.g. `C:\ASPKI.log`.

Signature Validation Rev Check (CRL) is another feature where the checking can occur both during creating of signature and signature validation on both the signing certificate as well as for the certificate associated with any revocation check responses. The feature has several options available

but one worth mentioning is to specify a LDAP server to query for CRLs. Additionally, querying an LDAP server might result in poor application performance if the quality of network connection is weak and the number of directories to search is too high. The attribute sLDAP has a data type as a string, the LDAP server gets CRLs from www.ldap.com [46].

2.13 Related work

This section describes some previous related work by students regarding digital signatures, public key infrastructures, and digital certificates.

2.13.1 Comparative study of digital signature usage in developed and developing countries

Jayakumar Thangavel [47] wrote about the use of digital signature in Sweden and India. His thesis compared the difference in how the two countries distributed certificates and how easy or hard it was to obtain a certificate. The study described the steps necessary for a citizen to obtain a certificate and thereby be able to applications to digitally sign documents and to verify their identity in order to utilizing an application's features. The widespread usage of Internet has had a huge impact on the adoption of digital signatures. This is as expected, as without an internet connection a user would not utilize digital signatures as they could not easily forward digitally signed documents to others.

2.13.2 Recommendations when implement PKI's

Johan Andersson [48] wrote a thesis about public key infrastructures, how PKI works, why PKI systems are needed, and what security aspects companies and individuals should reflect on before implementing a PKI. His work gives a good understanding of what security risks PKIs have, but also describes the benefits of utilizing PKIs. A functional PKI provides the conditions for wide spread utilization of digital certificates and digital signatures. His thesis excluded deeper reflection on digital signatures.

2.13.3 PDF of student transcripts

KTH (and several other Swedish universities) provide students the ability to get a transcript showing their grade, registrations, etc. via a web service. Additionally, these transcripts can be verified for up to 6 months after being produced. This service is well used but lack some security aspects. Today anyone logged into KTH intranet has the possibility to falsify another student's transcript by typing the corresponding personal code number from the certain student. With the usage of digital signatures, instead of verification codes and password protection of the .pdf file, these transcripts would have a higher security level as well higher credibility. This service is described at: <https://www.kth.se/en/student/program/intyg/intyg-1.322667>.

2.13.4 University use of digital signatures

A number of universities have adopted Adobe Acrobat and Adobe Sign for digital signatures, these include: Philadelphia University [49], Pepperdine University [50], Pace University [51], The University of Arizona [52], and University of Georgia [53]. Their implementation of Adobe's techniques has helped each university in different ways, but generally it has improved their managing of documents, speed up their processing, increased productivity, accelerated workflows, reduced paper work and archiving, and created greater mobile availability of documents.

2.14 Summary

To summarize, this chapter presented essential background information, described how cryptography is utilized in almost every step, from communicating between parties to signing documents. The use of private & public keys and a valid certificate signed by trusted authorities are fundamental to obtain a high level of security. The chapter described standards for secure message transfers and personal digital IDs to preserve a high level of trust and security. The fundamental technique in Adobe Acrobat Reader DC was briefly described.

3 Methodology

The purpose of this chapter is to provide an overview of the research method used in this thesis. Section 3.1 describes the research process. Section 3.2 focuses on the data collection techniques used for this research. Section 3.3 describes the experimental design. Section 3.4 explains the techniques used to evaluate the reliability and validity of the data collected. Section 3.5 describes the method used for the data analysis. Finally, Section 3.6 describes the framework selected to evaluate the proposed solution.

3.1 Research Process

This thesis will solve the current problem with KTH's use of paper forms by implementing digital signatures. The thesis will mostly focus on a specific form, see Appendix A, and the work will implement an already known technology from Adobe. Adobe's technique is considered to be one of the most widely utilized digital signature methods, its security level is sufficient, and therefore it seems to be a suitable tool for KTH to implement in their own administrative process to improve the efficiency of using application forms. The research process is split into three phases (as described below).

3.1.1 Phase 1: Thorough information gathering phase

This phase concerns the gathering of information and research necessary to know how to modify the current KTH user database (as realized by LDAP) and how to implement Adobe's digital signing via Adobe Acrobat Reader DC. This phase includes reading reports and articles written by vendors, students, faculty, and others.

3.1.2 Phase 2: Modify existing settings

This phase includes defining the preparatory steps necessary to implement Adobe's technique. Specifically it includes (1) identifying which certificates faculty, students, and staff should utilize, (2) how users should enroll in the system, (3) making entries in the LDAP database with the certificate(s) for each of the enrolled users, (4) defining the functionality of how these certificates are fetched and used by the user via an application such as Adobe Acrobat Reader DC, (5) revoking access by specific users, (6) revoking specific certificates, and (7) how to verify the signature on a document. If time is available, the use of timestamps will also be investigated.

3.1.3 Phase 3: Implement Adobe's technique as utilized in Acrobat Reader DC

The last phase is the implementation and evaluation of the proposed solution.

3.2 Data Collection

As the scale of this project is designed simply as a demonstrator, only a small number of users will be enrolled. In addition, rather than integrate the certificates into the actual KTH LDAP database a separate test LDAP database will be implemented.

3.2.1 Sampling

At least one test user from each of the following classes of users will be asked to participate, such that at least one user from each of the classes does participate in the demonstration. These classes of users are: student, faculty, and educational administrative staff (IT administrator will be mentioned but not included).

3.2.2 Sample Size

As described in the previous subsection, only a minimal number of users need to be involved in the demonstration. However, a few more samples will be taken from students as the initial part of the signing process is essential to analyze. Test users was determined with some criteria: (1) be a student at KTH, (2) between 19 and 30 years old, and (3) manage both Swedish and English in speech and writing.

3.2.3 Target Population

The real target population is all students at KTH (roughly 12,000 students), all of the teaching faculty (roughly 1,500 faculty members), and all of the administrative users (roughly 1,300 staff members).

3.3 Experimental Design and Planned Measurements

This section briefly describes the design of the test and the planned measurements. The test environment is described in Section 3.3.1 and the hardware and software are described in Section 3.3.2.

3.3.1 Test Environment

Initially each of the users will generate self-signed certificates. These certificates will be added to the test LDAP database.

A sample form, see Appendix B, will be generated for a fictional student registering for a thesis project. Each of the users participating in the demonstration will be asked to do their step in the processing of this sample form. These steps (the demonstration workflow) are:

1. The initially filled out form will be signed by the student and sent as an e-mail attachment to the participating administrator.
2. This administrator will check that the fictional student would be allowed to start a thesis project; if so, they will complete their part of the form and then sign the form and send it as an e-mail attachment to the participating faculty member.
3. This faculty member will complete their part of the form and then sign the form. They will send this completed form back to the administrator who will record the relevant information from the file into a database (or simply enter the relevant data into a spread sheet for the purposes of the demonstration) and then file the completed form.

3.3.2 Hardware/Software to be used

A Dell Optiplex 755 computer located in professor Maguire's lab will be used to host the demonstration LDAP server.

A self-signed certificate will be created for each of the users and be added to the LDAP server. The certificate will include some personal information about the user and the public key. The private key will be stored on the user's local computer or on an external device (USB).

The self-signed certificate can be exported from Acrobat DC in three different formats: .cer, .fdf and, .p7c. Depending on which program that connects to the LDAP server the certificate attribute might differ. In this case, with Acrobat DC, the LDAP attribute will be *userCertificate;binary* and the exported self-signed certificate will be in the format of .cer. The attribute *userCertificate;binary* stores the certificate as binary and contain personal information about the user such as: name, company, e-mail, country, etc, as well the public key.

Each user will use their usual computer running Adobe Acrobat DC to read, fill-in, and sign the form. Each of these users will use their normal mail program to send the data via the KTH mail system to the relevant next participant.

Check if the software can be configured to use the demonstration LDAP server – alternatively a bootstrapping program may be needed to fetch the certificates from the LDAP server to make them available to the software.

3.4 Assessing Reliability and Validity of the data collected

The signed document will be verified by each person who processes the document to ensure that it has been signed by the correct person in the previous step. The final document will be checked to ensure that all three signatures are valid.

It may be interesting to measure:

1. The time to set up a new enrollee with software, configuration, and initial user training.
2. The time required for each step of the processing – split into verifying, reading, decision making, and signing.

3.4.1 Reliability

Reliability will be limited as I will only measure the time for the various actions for one person from educational administrator and faculty member. The actions from students will have higher reliability than the previous case. However, the small number of participants is remarkably low compare to all thousands of students at KTH.

3.4.2 Validity

Validity will be ensured by using a XXX model YYY stopwatch for the timing of human actions and using the computer's internal high precision clock for timing of programmatic actions. The stopwatch's timing will be compared to that of a computer that is synchronized with KTH's network time protocol (NTP) computer. The computer's timing will be compared to N consecutive queries to the NTP each of which will be locally timestamped. This will enable a comparison of the relative difference in local times with the relative difference of the NTP server's times.

3.5 Planned Analysis of data

Data will be collected and analyzed with respect to the factors listed in Section 3.3 and Section 3.4. In this thesis the four different types of target users are: student, faculty member, educational administrator, and IT administrator. Depending on the type of user it is expected that the timing of these user's different action will differ. The first time to measure, in item 1, is how long time it takes to set up a new enrollee in the system with proper software, correct configuration, and good initial user training. Students, faculty members, and educational administrator are pretty straight forward to set up and therefore it should not take a long time to collect these measurements.

It is anticipated that the IT administrator will take longer time to perform their actions, as the IT administrator has greater authority and hence they must use great care in making any changes to the university's LDAP configuration – as all of the relevant security and functionality aspects need to be considered. In addition, the actions of the IT administrator could be split into smaller actions – corresponding to the specific authority of the individual administrator, for example one might be IT administrator concerned about the LDAP server and its configuration, another IT administrator concerned with certificates utilized by members of KTH, and a third IT administrator might be concerned with the process of setting up new users. The splitting of such IT administrative responsibilities is common for most large systems at KTH and other institutions with several thousand employees and many thousands of students.

The next time measurement, item 2, is how much time each process requires to be finished. As mentioned in Section 3.4 the processing can be divided into: verifying, reading, decision making, and signing. The first of these steps, verifying as document's signature may have a process time from milliseconds to a few seconds. This depends on how long it takes to check the signature validation against the CRL (unless the CRL is stored locally). The verification should not differ depending on which user has signed the document, as each user is expected to have the same basic type of certificate and the processing is expected to take place under similar conditions (i.e., one reasonably fast computers connected to a high speed network). The next step in the processing is the time required to read the document. This will differ depending on the document's length, complexity, and importance. Thus this step can be from seconds to many minutes. As the different users have different levels of experience and abilities regarding to their reading speed; therefore, this time is expected to vary depending on which user reads the document. For example, an educational administrator who processes hundreds of thesis project forms every year will require less time to process such a form that someone who only processes one such form per year. Next step in the process is decision making, how long a user takes to make a decision before signing the document or deciding not to sign it. The time to make a decision is expected to take longer than reading and will differ between users and may even differ depending upon who submitted the form. The final step is signing. This step is pretty straight forward as the user signs with their own self-signed certificate.

3.6 Evaluation framework

The proposed solution, create guidelines for KTH administrators to implement digital signatures, will most likely generate a faster signing process when a student applies for a thesis project compared to the current process. The method used to collect data and measure time for each type of user described in Section 3.4 and Section 3.5 will be the framework when students, educational administrator, and faculty member digitally sign a sample form (see Appendix B). They will also receive an attached text file (see Appendix C) with basic instructions to successfully create a digital ID and to sign the sample form. These actions will be described in detail in next Chapter.

4 Implementation and Result

The purpose of this chapter is to describe and explain the steps necessary to implement Adobe's digital signing technique in the context of this thesis project. This chapter also describes the decisions that were made and why they are the most suitable decisions for deployment at KTH. The emphasis in the evaluation will be how the proposed digital signature technique facilitates the complete workflow for application forms. The result from the collected data will be described in Section 4.3.

4.1 Creating digital ID in Acrobat Reader

To illustrate the signing process when using digital signatures, the first step is to create a digital ID using Adobe's Acrobat Reader DC. Creating a digital ID is easily done by the following steps: press edit, preferences, signatures, identities & trusted certificates, and finally add digital ID. In the process of adding or creating a digital ID you must either use an already created digital ID stored in a file, on a roaming server, or on a device connected to the computer *or* create a new digital ID. Creating a new digital ID ultimately produces a PKCS#12 digital ID file format (password-protected file). The next step in the process is to enter the individual user's information such as name, email address, company name, and country; choose the RSA key algorithm (with 1024 or 2048 bits); and indicate what purpose this digital ID will serve. The final step is to store the new digital ID in a file (or device) and create a password for this ID, this password is required every time the ID is utilized. The ID is created as a .pfx file, e.g. PontusAlexanderEngström.pfx. It will typically be placed in a directory such as C:\Users\Pontus\AppData\Roaming\Adobe\Acrobat\DC\Security. This self-signed certificate issued by Adobe (see Section 2.8.3) can be both imported and exported to other users for the purpose of validate documents signed by that specific ID.

This ID will be created for all members at KTH. Students, teachers, faculty members, staff, and so on. They all will generate a self-signed certificate created in Acrobat DC to be able to utilize the digital signature feature.

4.2 LDAP

With a self-signed certificate, it is now possible to import it to KTH's current LDAP database that contains information about students, faculty, and staff. Figure 4-1 and Figure 4-2 shows the attributes and objects stored in the LDAP for an example student and faculty member (respectively). This syntax is briefly described in Section 2.7.4.

```

1 # Pontus Engstr\C3\B6m (poneng), Addressbook, kth.se
2 dn:: Y249UG9udHVzIEVuZ3N0csO2bSAocG9uZ2W5nKSxvdT1BZGRyZ2XNzYm9vayxkYz1rdGgsZGM9c
3   2U=
4 objectClass: top
5 objectClass: ugAuxUser
6 objectClass: ugAuxObject
7 objectClass: inetOrgPerson
8 objectClass: organizationalPerson
9 objectClass: person
10 objectClass: eduPerson
11 objectClass: norEduPerson
12 ugUsername: poneng
13 ugKthid: ulvoxtcp
14 givenName: Pontus
15 sn:: RW5nc3Ryw7Zt
16 ugClass: user
17 displayName:: UG9udHVzIEVuZ3N0csO2bQ==
18 mail: poneng@kth.se
19 cn:: UG9udHVzIEVuZ3N0csO2bSAocG9uZ2W5nKQ==
20 labeledURI: https://www.kth.se/profile/ulvoxtcp/
21 eduPersonPrincipalName: ulvoxtcp@kth.se

```

Figure 4-1: Example of a student in the LDAP data base

```

1 # Gerald Q Maguire Jr (maguire), Addressbook, kth.se
2 dn: cn=Gerald Q Maguire Jr (maguire),ou=Addressbook,dc=kth,dc=se
3 objectClass: top
4 objectClass: ugAuxUser
5 objectClass: ugAuxObject
6 objectClass: inetOrgPerson
7 objectClass: organizationalPerson
8 objectClass: person
9 objectClass: eduPerson
10 objectClass: norEduPerson
11 ugUsername: maguire
12 ugKthid: uld13i2c
13 ugClass: user
14 mail: maguire@kth.se
15 labeledURI: https://www.kth.se/profile/uld13i2c/
16 eduPersonPrincipalName: uld13i2c@kth.se
17 cn: Gerald Q Maguire Jr (maguire)
18 givenName: Gerald Q
19 sn: Maguire Jr
20 displayName: Gerald Q Maguire Jr

```

Figure 4-2: Example of a faculty member in the LDAP data base

The entry above was retrieved using the query:

```
ldapsearch -x -H ldaps://ldap-master.sys.kth.se -b ou=Addressbook,dc=kth,dc=se
ugUsername=maguire
```

As displayed in the above figures no certificate attributes exists, therefore the addition of *userCertificate;binary* shouldn't be a problem. The IT administrators will of course look deeper into the security aspects before adding certificates to each user in KTH's intranet.

The enrollment process is created when a user (student or employee) tries to log in to KTH's webpage with its username and password. When the log in process is complete the LDAP server will know which user it is and therefore send the certificate assigned to that user (when it is needed). When it is fetched the user can start to sign documents. Note that the user can sign documents digitally without the fetched certificate (only private key is needed to sign) but without a certificate, which include a public key, no recipient can validate the signature.

Creating entries in the existing LDAP database will be similar for each enrolled user. The entry will be below *ou=Addressbook, dc=KTH, dc=se* (which is already created). Below *ou=Addressbook* there exist a bunch of common names (cn), which is the user itself. This is the valid path to insert the

attribute *userCertificate;binary* and store one certificate per user. This thesis does not include multiple certificates for one user with different employment, because of the lack of time.

The functionality process of how to fetch and use these certificates in an application such as Adobe Acrobat Reader DC can be described as follow.

With an LDAP server containing certificates it is possible to fetch them for signature utilization. When a .pdf is opened in Acrobat DC the users certificate will be fetched through these steps: run Adobe Acrobat Reader DC > edit > preferences > signatures > identities & trusted certificates > add ID > my existing digital ID from: a roaming digital ID accessed via a server > enter the URL from the LDAP server. The URL for KTH's LDAP server should be *ldaps://kth.se:636*.

When a user has fetched a certificate from the LDAP server and has their private key stored on either their local computer or on an external device such as USB, the availability to digitally sign documents and to validate other signatures are now granted. If the certificate becomes invalid (time period expires, a student graduates, etc.) it will be revoked and will not be used to sign documents anymore. When this occurs their certificate is stored in a CRL and will never be removed, the user must create a new valid certificate. Acrobat DC has by default a configuration that enables checks against a list of excluded certificates during the validation process. Additionally, the revoking process will be the same for all users in KTH's intranet since all certificates are the same and treated equally.

The first step to verify and validate a signature in the interaction is when a student signs the application form for a thesis project and sends it to the educational administrator. Before the educational administrator sign the form the students signature has to be investigated, if it is valid or not, then the administrator can search into the database system if the student has enough credits to start a thesis project. If so the administrator signs the document and sends it to a faculty member who will help the student during the thesis project. Before the faculty member sign the document both signatures from student and educational administrator will be investigated by the server to prove their validity. When the document obtains all three signatures the faculty member sends it back to the educational administrator who adds the course into LADOK so the student gets registered. Note that every time a user has sign the document it will get checked before next part will sign it. Additionally, timestamps were not investigated because of lack of time. This feature will be mentioned in Section 6.3 for future work.

4.3 Result from the Collected data

During the implementation stage some data has been collected. It was divided in two parts, both creation of a digital ID and reading and signing a document. These measurements will be described below.

In the first stage of the sample form a student has to create a digital ID, see Figure 4-3. The data population was based on the inclusion criteria mentioned in Section 3.2.2. This task was sent out to 15 different students where they had to clock their time during their creation of a digital ID. They also received a text file "readme.txt" with basic instructions of how to create a digital ID, see Appendix C. This text file was created due to the probability that KTH creates their own for students and personnel when they want to create their first digital ID.

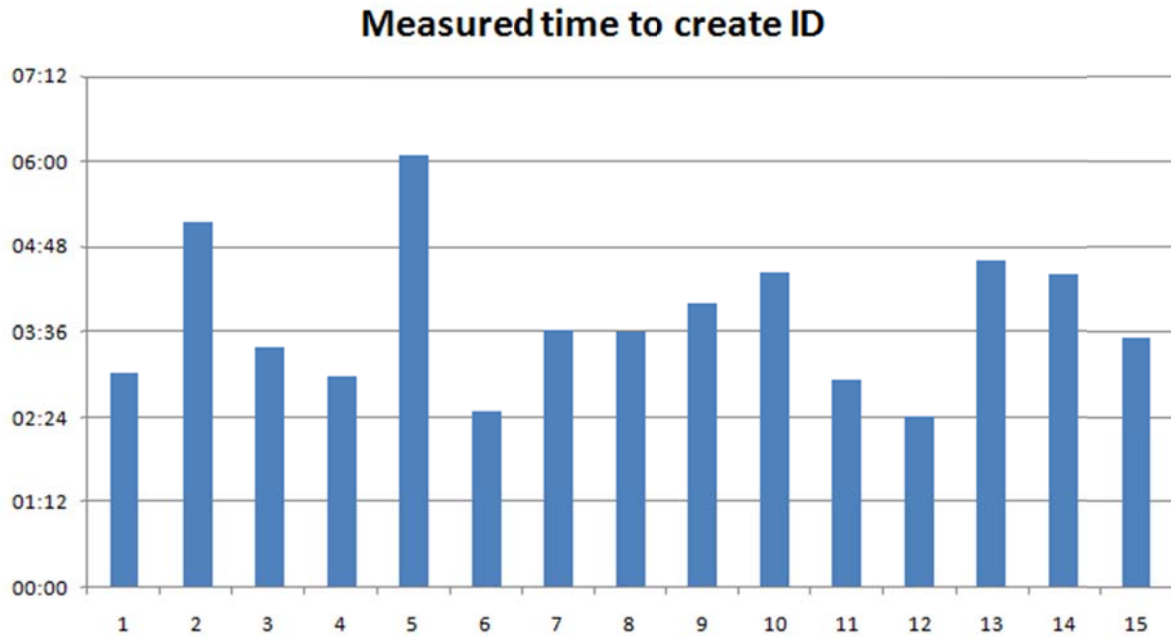


Figure 4-3 Measured time (in minutes: seconds) with 15 different students to create a new digital ID via Acrobat

When a student has successfully created a digital ID they will most likely sign the document. These documents might differ but in this thesis it will only be the basic sample form with no additionally description than the headline "Application form for thesis project" and three parts to sign the document. Therefore, certain time factors will be more or less important in the process of signing (in this case reading). These time factors are verifying, reading, decision making, and signing. Figure 4-4 displays the measured time for all students when they signed a document. This signing process included: reading the document, decision making to either sign or not sign the document, and to put the signature on the document. The verifying phase will be used for educational administrators and faculty members when they have to verify all signatures on the document. When a student wants to start a thesis project no signature will appear on the document besides their own, therefore no further verifying for the student except to verify that they sign the correct document.

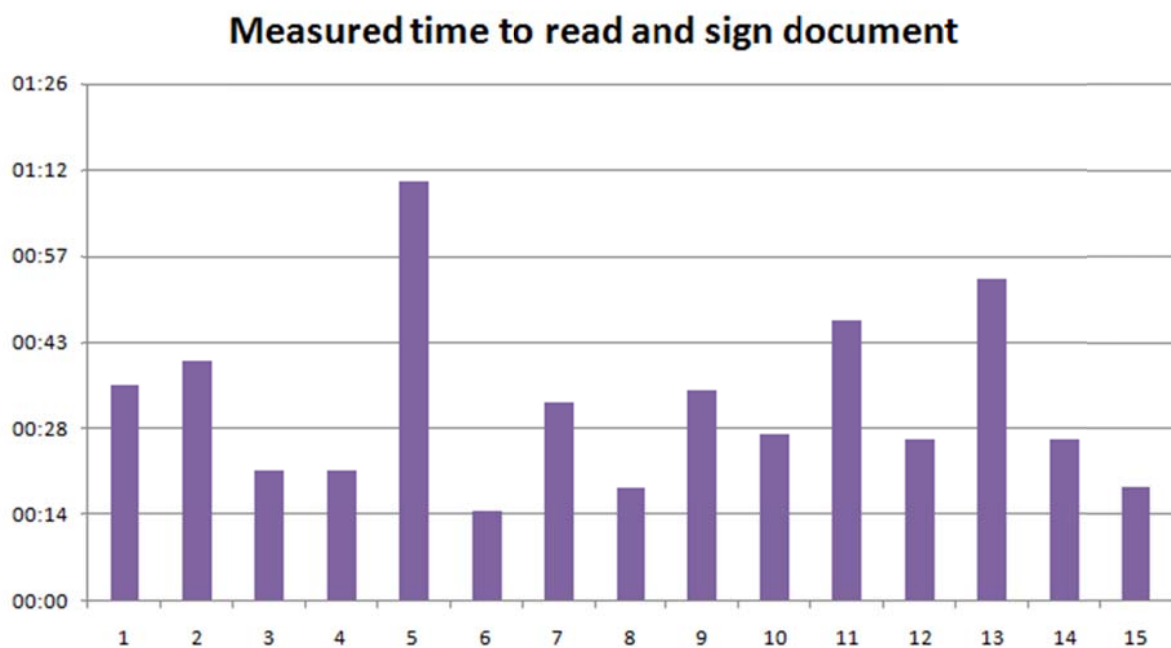


Figure 4-4 Measured time (in minutes: seconds) with 15 different students to read and sign a document

With both the creation and signing measured it can also be interesting to see their data added together. Figure 4-5 displays the measured time to both create a new digital ID, and to sign a document.

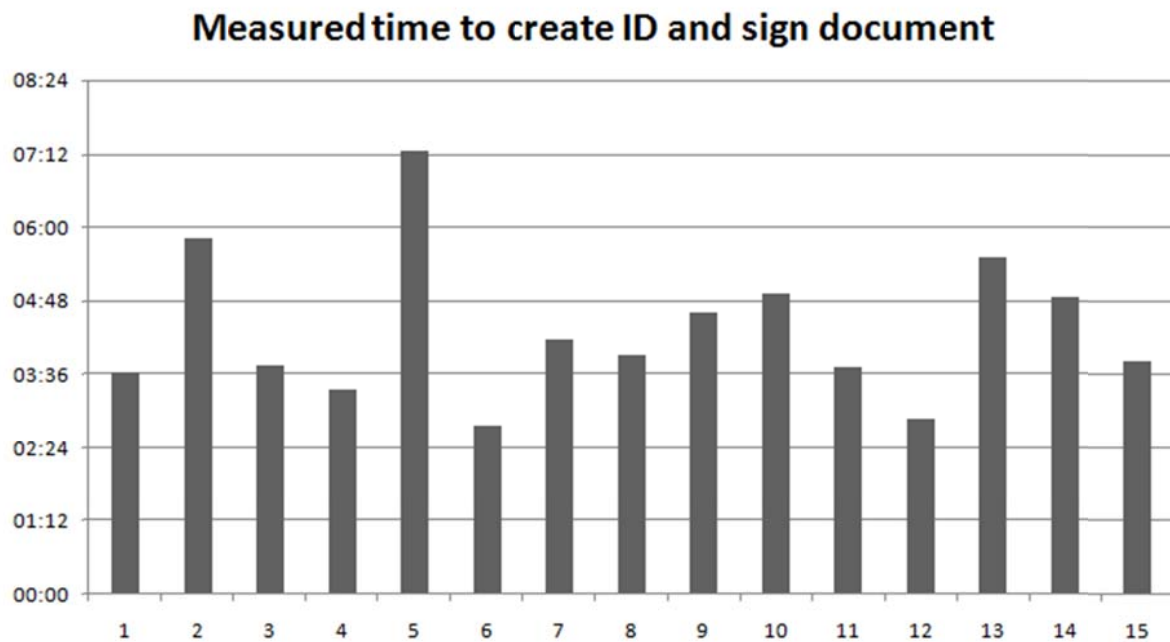


Figure 4-5 Measured time (in minutes: seconds) with 15 different students to create a ID and to sign a document

The last thing to do for a student is to send this document to the educational administrator. This process will also differ consider that each individual express themselves differently through email conversations.

The data collected from the measured time it took each student to send an email with the attached document and some text in the mail itself was collected by asking them through Skype, Facebook or other social media. Figure 4-6 displays the time needed for certain students to fulfill their part of the signing process, include the time to write an email with the attached signed document.

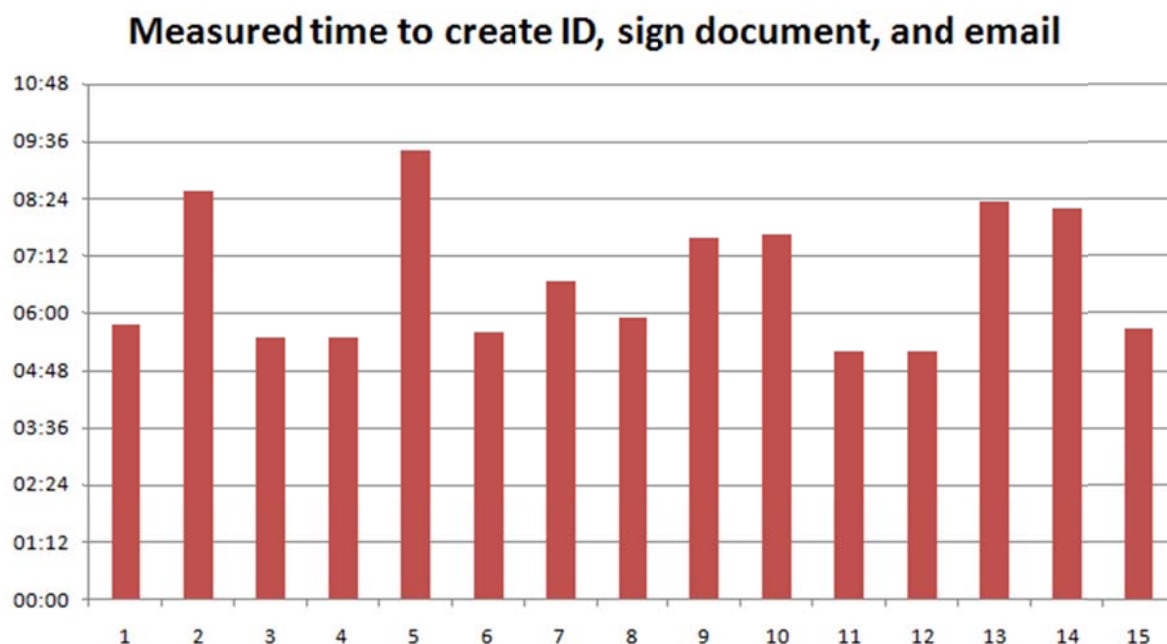


Figure 4-6 Measured time (in minutes: seconds) to create ID, sign document, and to email the next part to sign the application form

The next stage in the application form process is when the educational administrator retrieves the signed document and determines if the student is authorized to begin a thesis project. The educational administrator has to include the time needed to check a student's credits before determine to sign or not sign the document, then to the same steps as the student: 1) create digital ID and, 2) read and sign document. Unfortunately, the educational administrator failed to return data in 4 weeks.

The last stage in the application form process is when the document is signed by a student and an educational administrator. The signed document is retrieved to the faculty member by email from the educational administrator who will do their part to this final stage in the signing process. The faculty member in this thesis is Gerald Q. Maguire Jr. who clocked his time when creating a new digital ID and signing the document to: 5:58 (minutes: seconds) where most of his time (4 minutes) due to avoid the usage of an already existing digital ID. When he had read and decided to create a new digital ID and sign the document, it took 1:58. He approximated it would take another 2 minutes to attach and send the application form with all signatures back to the educational administrator. The administrator will record relevant information from the file into a database and then file the completed form.

The data collected of the time needed by the 15 students to create a digital ID seems to be a normal distribution, as shown in Figure 4-7. Only by studying the chart it is obvious that the mean is close to 230 seconds, minimum near 100 seconds and maximum near 400 seconds.

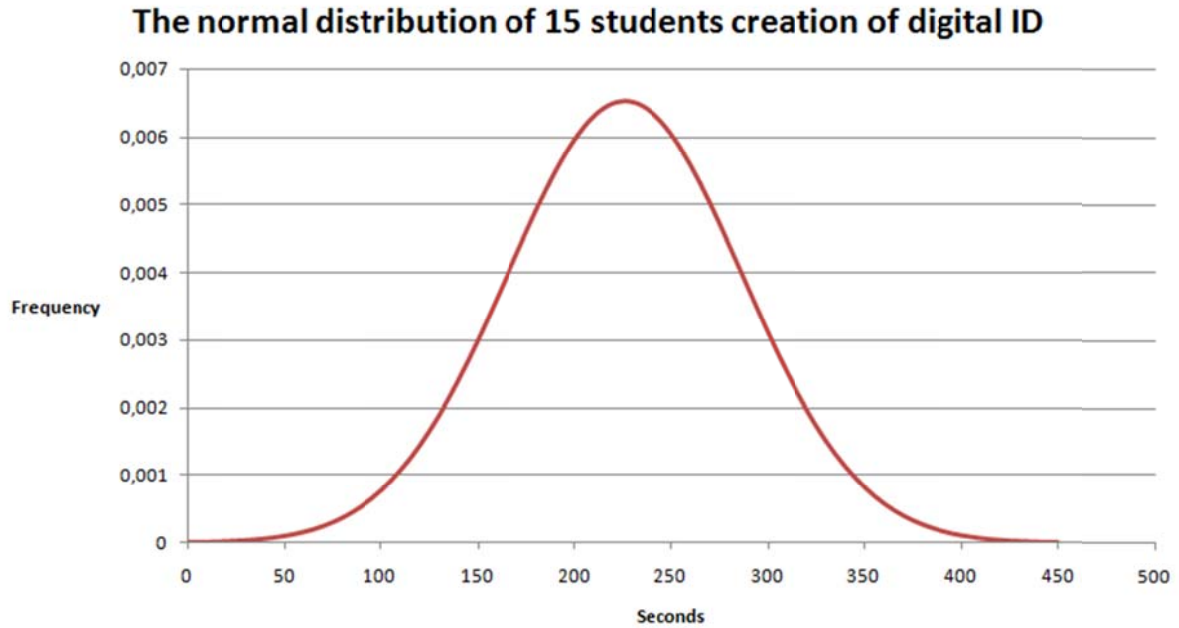


Figure 4-7 The normal distribution of 15 students creation of digital ID displayed as a bell curve

These collected data are a small sample but can be described as a normal distribution with a 95% confidence interval. Firstly, the mean value and the standard deviation are calculated:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

$$\bar{x} \approx 226,73$$

Assuming that X is a random variable with the mean value μ :

$$E[X] = \mu$$

The variable E is the expected value of X , then the standard deviation is:

$$\sigma = \sqrt{E[(X - \mu)^2]} = \sqrt{E[X^2] - (E[X])^2}$$

Since this is a case when X takes random values from a finite set where each value has the same probability, the standard deviation is:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

$$\sigma = \sqrt{\frac{1}{15} \sum_{i=1}^{15} ((181 - 226,73)^2 + (309 - 226,73)^2 + \dots + (211 - 226,73)^2)} \approx 61,14$$

The standard deviation displays how much the collected data departs from the mean. In this case it will have an interval: 165,59 – 287,87.

With these calculations it is possible to determine the normal distribution (which is shown in Figure 4-7). The random variable X is normally distributed with parameters μ and $\sigma > 0$ if

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

for all x . The code notation for normal distribution X is $N(\mu, \sigma)$. The distribution is symmetrical around μ therefore is μ the expected value: $E[X] = \mu$. Additionally, $D(X) = \sigma$.

If X is $N(\mu, \sigma)$ then:

$$Z = \frac{X - \mu}{\sigma}$$

has the expected value $E[Z] = E\left[\frac{X-\mu}{\sigma}\right]$

It can be interesting to calculate the expected outcome between 200 and 300.

The already calculated values of $\mu \approx 227$ and $\sigma \approx 61$, gives the probability of:

$$P(200 < X < 300) = P(200 - 227 < X - \mu < 300 - 227) = P\left(\frac{200 - 227}{61} < \frac{X - \mu}{\sigma} < \frac{300 - 227}{61}\right)$$

Since $Z = \frac{x-\mu}{\sigma}$, $\frac{200-227}{61} = -0,44$ and $\frac{300-227}{61} = 1,2$ which results in:

$$P(200 < X < 300) = P(-0,44 < Z < 1,2)$$

$$P(-0,44 < Z < 1,2) = P(Z < 1,2) - P(Z < -0,44)$$

Using the standard normal table* the values can be determined as:

$$P(Z < 1,2) = 0,8849$$

$$P(Z < -0,44) = 1 - P(Z < 0,44)$$

$$P(Z < 0,44) = 0,67$$

$$P(Z < -0,44) = 1 - P(Z < 0,44) = 1 - 0,67 = 0,33$$

This results in the probability of: $P(-0,44 < Z < 1,2) \approx 0,55$

The calculation tells us that that nearly 55% of all possible outcomes are between 200 and 300 seconds, Figure 4-8 displays this calculation.

* Can be found at: <http://www.maths.lth.se/matstat/kurser/tabeller/tabeller.pdf>

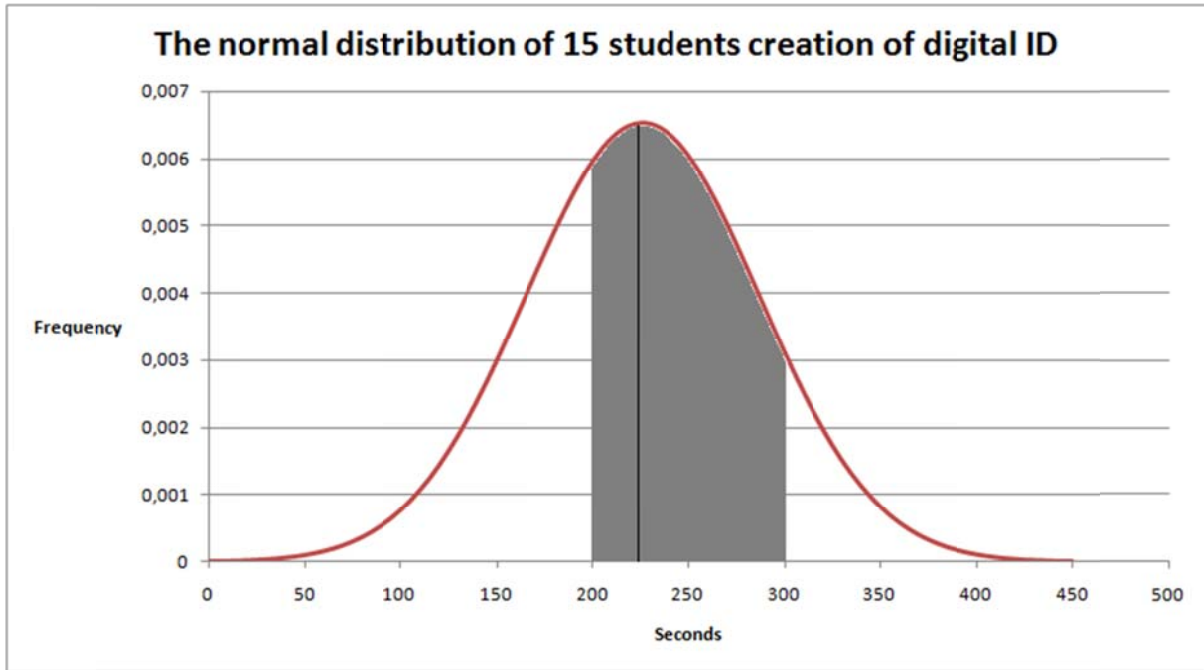


Figure 4-8 The probability that $200 < X < 300$ is equal to the grey area under the curve (black line is mean value)

The calculation of a 95% confidence interval might be relevant in this case.

As mentioned before, $\mu \approx 227$ and $\sigma \approx 61$.

With help of the standard normal table for each possibility α we can determine the quartile $\lambda_{\alpha/2}$ so that

$$P = \left(\left| \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \right| \leq \lambda_{\alpha/2} \right) = 1 - \alpha$$

In other words, with probability $1 - \alpha$ is

$$-\lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \bar{X} - \mu \leq \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

or

$$\bar{X} - \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

This is a symmetric confidence interval for the expected value, with confidence $1 - \alpha$. With the confidence $1 - \alpha = 0,95$ it entails $t_{\alpha/2} = t_{0,025} = 1,96$. This data was retrieved from the table of quartile for normal distribution (the same table shown in previous calculation).

The observed interval will be:

$$\begin{aligned} \bar{X} - \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}} &= 227 - \left(1,96 \times \frac{61}{\sqrt{15}} \right) \approx 196,13 \\ \bar{X} + \lambda_{\alpha/2} \frac{\sigma}{\sqrt{n}} &= 227 + \left(1,96 \times \frac{61}{\sqrt{15}} \right) \approx 257,87 \end{aligned}$$

Results in: $196,13 \leq \mu \leq 257,87$

The result indicates that with 95% certainty the mean value of the whole population will be in the interval of $196,13 \leq \mu \leq 257,87$.

These calculations are based upon the data collected only from 15 students and their time to successfully create a digital ID. The same calculation can be performed on all collected data. These

data calculations will not be described in detail as above example, but they are calculated with the same equations and summarized up in Table 4:1. All equations are covered in the book by Gunnar Blom, Jan Enger, Gunnar Englund, Jan Grandell, and Lars Holst, *Sannolikhets teori och statistik teori med tillämpningar* [54].

Table 4:1: This table displays the mean value, the standard deviation, and the 95% confidence interval of the data collected from 15 students

	Create ID	Sign document	Create ID and sign document	Create, sign and email
\bar{X}	227	33	260	403
μ	61	15	72	86
95% interval	196 - 258	25 - 41	223 - 296	360 - 447

5 Analysis

The purpose of this chapter is to analyze and evaluate what I have done, what my metrics are, and to analyze my obtained data and proposed solution. I will also analyze if I met my goals, the goals I had when I started the project, and discuss other aspects regarding my thesis work and digital signatures.

5.1 Major results

The major result of this thesis project is a demonstration of an implementation of digital signatures as used to process an example form through its typical series of signatures. In Chapter 4 it was described how the implementation will work as well benefit the workflow from previous signature method. With the usage of digital signature, it will not only benefit students with their waiting to start their thesis project, but also ease the communication for educational administrator who can do the most of its work through emails. This new technique will also facilitate faculty members and the less printed papers which spares the environment.

5.2 Reliability Analysis

The most of the collected data are limited due to the small number of participants. Only one person of the type educational administrator and faculty member are observed and therefore only gives a small scope of the timed needed to accomplish each step in the signing process. Unfortunately, no data was provided by educational administrators.

The data collected from the faculty member, Professor Q. Maguire Jr., is only one sample. Furthermore, the time taken would most likely be lower if he does the same procedure again, considering that it took approximately 4 minutes due to avoid the use of an already existing digital ID. With another test, it would presumably be lower by 3-4 minutes.

The only part that is somewhat reliable is the data collected from the 15 students. These data differ from person to person, most likely because of how each student approached their task (read, decision making, create ID, sign document, and send email). The complexity of the document is not particularly high, therefore some factors were hard to determine (for example, the decision making for students - since they had *already* made a decision to start a thesis project was outside the period that was measured) and was coupled with other factors. However, decision making is still a factor for educational administrators and faculty members since they both have to make a decision of whether to approve the request by the student who signed the application form. The time taken to read the document will also differ depending on the reader and their familiarity with this type of document. In this thesis the sample form is simply twelve lines, where each type of person has to fill in four lines (first name, last name, date and, the digital signature, see Appendix B). The form does not have additional text that describes what is required of a student who applies to start a thesis project or if the student wants a rating scale of F/P or A-F, it only has a title "Application form for thesis project". All of this other information is provided in other documents (which the examiner has to consult when meeting with the student for the method and planning meeting or which the education administrator has to access from other documents and databases). However, the conclusion is that reading does not take a long time for any type of user. If the form was re-designed (which is likely in the future) the time taken by a student to read the application form will presumably be comparable to this sample form. Educational administrators who process these forms weekly do not require much time to read the form as they only look who is applying and then check to see if they are eligible to apply (which requires checking a data base to see if the prerequisites have been completed and if the student has enough points to start such a thesis project). The same applies for faculty members, although reading time might differ depending on how many thesis projects the faculty member has previously been

involved in (some faculty members have a lot of thesis projects ongoing each year and some have only a few). The process of creating an ID is identical for all parties since they use the same software and the same type of certificate, the same applies to document signing. The last factor, emailing the signed form, will differ depending on how each person chooses to communicate. This is hard to evaluate, but from the data collected from students the conclusion is that the email process is quite similar for all students (usually it takes 2-3 minutes to attach the signed document and write some text).

5.3 Validity Analysis

The validity of the collected data was hard to determine. Each involved user in this thesis measured their own time, therefore some data might be higher than anticipated. Note that this thesis project aims to implement digital signature for a specific application form due to the long time delay for students (one-two weeks). If a few data samples have some seconds or maybe a few minutes that are not supposed to be measured (a test user can easily be distracted when they do this signing procedure at home) it will not affect the whole process because the impact will be minimal. However, it is important to mention the low validity of these data and if these collected data should be performed again it is highly recommended to have some kind of time unit (a test user that only focuses on clocking the actual time needed for another user to fulfill a signing process, or an application that clocks the programmatic actions).

5.4 Discussion

The data collected from 15 students are a small sample size in this context of thesis project, although this is more samples than were available for the other types of users. In Section 4.3 it is quite obvious that the analysis outcome has a wide interval. With more participants the data would have been more reliable. The variation of the 95% confidence interval is a typical indication of a small sample considering the difference from the mean value. I expect that with a larger number of samples most of the data would be closer to the mean, but there would still be a wide range of results due to the variation between users. Note that 15 students of totally 12.000 students at KTH is a remarkably low sample size. Individuals have different computer experience and it might reflect the data collected, if a new test would be performed, with hundreds of students, the result will be reliable but the interval could be similar as this thesis project demonstrates.

The collected data, from students, visualize a fast signature process (6-7 minutes). Although no sample was received from educational administrators, digital signature is still considered to be greatly faster than the current signature process at KTH. Even if the educational administrator took one or two days to complete their part of the signature process, it would still be faster than to wait two weeks for each student to eventually start their thesis project. The faculty member could also take one or two days before completing their part of the process, but this would not change the fact that digital signature is way more efficient. Even without a sample from educational administrator, a logical theory could be: with digital signatures the complete signing process could be done during *one* single day.

Other aspects than time measurements could have been done. For example, if test users found digital signatures as a user friendly tool and if they would like to use it in other scenarios than the one described in this thesis project. Another question could have been if test users experienced the creation of ID as a difficult process, if so maybe an application would be developed to facilitate the creation of ID's (with a few clicks).

A major aspect that has not been truly discussed is the security. The security regarding how Adobe handles key pairs, hashes, signatures, etc., is not questioned, rather the utilization of self-signed certificates. It seems to be safe enough in the context of this thesis, but one aspect such as the enrollment process (how to initially make digital signatures available for users) might have some security issues (because anyone can create an ID with someone else's name). An example of how to

ensure a high security in the enrollment process could be; when a new student starts their first school period and receives their access card they should also meet an IT administrator where they together can create a digital ID for the student. This process is exactly as the process described in this thesis work. Note that the IT administrator is only there to support and to give guidelines to create an ID (the password created by the student is only for the student, **not** for the IT administrator). However, if the ID administrator somehow sees the typed password, the system can still be considered to have a high security because this technology is based on a *two-factor authentication system*, where a user must obtain both the digital ID **and** the password for that specific ID to make digital signatures. The administrator is also there to receive the student's certificate so it could be fetched from LDAP. It could be good for the student to sign a test document to get the knowledge of how to actually use the created ID. Another important aspect is that each student has to store their newly created ID on a computer or USB device. For efficiency, the computer used to create ID's will be a KTH computer and the created ID will **not** be stored on that specific computer. KTH has a lot of companies "headhunting" new examined students to eventually begin working with them, therefore it would not be a big problem to somehow acquire a few hundreds (or thousands) of USB devices (with small memory space) where the company who sponsor most likely will have their label or logotype on the USB device (for marketing). However, with these USB devices each student could store their ID on it and the security would be considered sufficient. Also note that before consulting with an IT administrator each student must somehow identify themselves (passport, driving license, etc.) to either the IT administrator or to someone else who can check their identity to a data base for new students.

This process could be replaced by a similar one that is already used, when students receives their log in credentials to access kth.se. These credentials are already generated for the student, but it is highly recommended (if not required) to change the password after a few days. This could be considered to be a lack of security since this whole thesis work is relying on each individual is the only part knowing of their private key (roaming ID's are outside the scope of this thesis work, but could be considered as a solution if this method was prioritized). If IT administrators also knew the password the system could be weak and forgeries might occur.

Another security aspect could be to store a CRL locally on the LDAP. This could facilitate the process for a user to check if a signature is valid or not. Usually people who already have initialized a communication have stored recipients certificate on Acrobat, to easily check if the signature matches with a corresponding certificate. The CRL will be added with certificate from students that loses their digital ID, ends their studies at KTH, or if students get a feeling that someone else is using their ID for signature purpose.

With a fully implemented and integrated digital signing technique, KTH has many opportunities to develop their usage of digital signatures in other areas. In some courses, students must sign a course form at the beginning of the course to gain access to specific material and to be registered for the course. With digital signatures these actions can be done even without attending the first lecture. Signing this form, to show an interest in taking the course, can be done at home or any other place. The overall savings by utilize digital signatures can reduce administration time, reduce delays for students (or staff), and reduce the amount of paper needed (having a positive environmental impact – not only because of the reduction in the use of the paper itself, but also due to the reduction in the transport of this paper and its archiving).

6 Conclusions and Future work

This last chapter gives the conclusions of the thesis project as well suggests some future work for other students or IT administrators at KTH if they want to develop a deeper understanding of digital signatures.

6.1 Conclusions

The goals of this thesis as defined in Section 1.4 were not all successfully accomplished. This thesis project created a simple prototype and offers suggested guidelines for KTH IT administrators if/when they implement digital signatures. The factors that were covered: multi-signing (by different users), expiration date of certificates and signatures, and other aspects for an invalid certificate were covered in Section 2.6.3. However, two factors were not covered: logs and timestamping through a TSA.

The two factors that were not covered are essential parts to have a high security when utilizing digital signatures. If timestamping through a TSA would be implemented no one can reject a signature. Logs would make it possible to identify failures in the workflow, e.g. a user can identify if someone has used their signature to sign document without the knowledge and approval from the originator. These techniques are important to implement if KTH want to utilize digital signatures in their work environment.

The major result from this thesis project, except the creation of guidelines to implement digital signatures, is the fast signature process utilizing digital signatures. Based on the data collection in Chapter 4 and the analysis of these data in Chapter 5, it is clearly not only useful but recommended to implement this technique.

The insights I have gained are not just all the knowledge of digital signatures and certificates, but also understanding of the continuing growth of the digital signature market. This technique is greatly underestimated, but achieving its full potential effectiveness is difficult. Documents (such as agreements) are essential for many companies and industries. Digital signatures not only ease the interactions between different parties, but also create a much more trustworthy and efficient working environment. The security seems to be higher than regular ink-written signatures.

For future students or other individuals who want to do research in this area, my suggestion is to start investigating basic cryptography and to understand its fundamentals. The other essential aspect to understand is why the use of digital signatures benefits all parties when compared to classic handwritten signatures.

If I were to do this thesis project again I would look deeper into the weakness of digital signatures and where they might occur (and if they might occur). In this thesis project only self-signed certificates are created, but with certificates issued by Certificate Authorities the reliability increases a lot. I would also look deeper into RSA- & DSA algorithms since their complexity interests me.

6.2 Limitations

The main limitation of this thesis project was due to its bounded time. It was hard to determine how much time each research area needed and how to obtain all facts necessary to fully describe a functional technique that KTH could take into account for their future work. Note that this thesis project is not guaranteed to be used as a template or guidance to the staff of KTH, they will most likely take other features into account before making any decisions regarding the utilization of digital signatures.

6.3 Future work

Some suggested future work based upon this thesis project includes:

- Continue to work with timestamps. This feature is essential to obtain higher security in digital signatures. This might be something for the IT administrators at KTH to look into deeper.
- Use digital signature on more documents than just the application form utilized in this thesis project.
- Utilize digital signatures in scenarios when students have compulsory attendance during the start of a new course. With digital signatures a student can sign the application sheet at home and does not need to attend the first class.
- Collect more data from educational administrators and faculty members. Use this data to determine a 95% confidence interval and compare the times to those calculated in Section 5.3.
- Student transcripts were mentioned in Section 2.13.3. To implement digital signature on these transcripts would improve the reliability of the transcript as well their security so they cannot be falsified so easily.

6.4 Reflections

The reflections worth mentioned regarding economic, social, environmental, and ethical aspects of my work include:

- Economic - my proposed solution does not cost a substantial amount of money since the technique is widely available in numerous products (including ones that are available free of charge). However, there will be costs associated with KTH's IT personnel to set such as system up and to operate it. Before KTH implements digital signatures administrators would most likely calculate the time to implement it and determine whether this is a high priority activity to implement in comparison with other tasks they have.
- Social - the usage of digital signature makes it harder to falsify an individual's signature. This technique would enrich the society, not only for KTH and other universities, it would enable more reliable document (such as agreements). The usage of digital signatures would decrease the necessity of having a meeting where each party makes a decision of whether to sign the agreement or not.
- Environmental - using digital signatures would decrease the use of printed paper, therefore saving the environment. With less paper being printed this would also result in an economic saving by eliminating the need to transport and store paper documents. This reduction in transport of papers results in less environmental emissions.
- Ethical - a major ethical concern is the need to securely store the user's private key (even when signing), as disclosure of this key could have serious negative effects on the user's personal integrity.

References

- [1] Richard J. Sullivan, 'Can Smart Cards Reduce Payments Fraud and Identity Theft?', *Federal Reserve Bank of Kansas City Economic Review*, vol. Third Quarter, pp. 35–62, 2008.
- [2] Jacquinot Consulting, Inc., 'Smart Card Applications', *CardWerk Smart Card Solutions*. [Online]. Available: http://www.cardwerk.com/smartcards/smartcard_applications.aspx. [Accessed: 22-Apr-2016]
- [3] Raheemah Abdulaleem, Len Kardon, Stephen Y. Chow, Jorge Contreras, and Linda Hamel, 'E-Commerce: An Introduction- Session 3: Transactions', *E-Commerce: An Introduction*, 30-May-2001. [Online]. Available: <http://cyber.law.harvard.edu/olds/ecommerce/transactions.html>. [Accessed: 22-Apr-2016]
- [4] Entrust Inc., 'Digital Signatures', *Entrust: Securing Digital Identities & Information*. [Online]. Available: <https://www.entrust.com/digital-signatures/>. [Accessed: 22-Apr-2016]
- [5] Jalal Feghhi, Jalil Feghhi, and Peter Williams, *Digital certificates: applied Internet security*. Reading, Mass: Addison-Wesley, 1999, ISBN: 978-0-201-30980-5.
- [6] Adobe Systems Incorporated, 'Electronic signatures, online e-signatures', *Adobe Acrobat DC*, 19-Apr-2016. [Online]. Available: <https://acrobat.adobe.com/us/en/how-to/electronic-signatures-online-e-signatures.html>. [Accessed: 22-Apr-2016]
- [7] Adobe Systems Incorporated, 'Certificate-based signatures', *Adobe Acrobat DC*. [Online]. Available: <https://helpx.adobe.com/acrobat/using/certificate-based-signatures.html>. [Accessed: 22-Apr-2016]
- [8] U.S. Department of Homeland Security, Office of Biometric Identity Management (OBIM), 'Federal Programs', *Biometrics.gov*. [Online]. Available: <http://www.biometrics.gov/ReferenceRoom/FederalPrograms.aspx>
- [9] Baran Topal, *Comparison of Methods of Single Sign-On: Post authentication methods in single sign on*, Master's thesis. Stockholm, Sweden: KTH Royal Institute of Technology, School of Information and Communication Technology, 2016, TRITA-ICT-EX-2016:14 [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-183144>
- [10] P.K. Yuen, *Practical cryptology and Web security*, vol. 2006. New York: Pearson Education LTD, ISBN: 978-0-321-26333-9.
- [11] Whitfield Diffie and Martin E. Hellman, 'New Directions in Cryptography', *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976. DOI: 10.1109/TIT.1976.1055638
- [12] R. L. Rivest, A. Shamir, and L. Adleman, 'A Method for Obtaining Digital Signatures and Public-key Cryptosystems', *Commun. ACM*, vol. 26, no. 1, pp. 96–99, Jan. 1983. DOI: 10.1145/357980.358017
- [13] SecMaker, 'Net iD Enterprise'. [Online]. Available: <https://www.secmaker.com/net-id/software/net-id-enterprise/>. [Accessed: 24-May-2016]
- [14] neXus group, 'neXus Personal Security Client'. [Online]. Available: <https://www.nexusgroup.com/en/topics/authentication-access-management/nexus-personal-security-client/>. [Accessed: 24-May-2016]
- [15] Finansiell ID-Teknik BID, 'BankID Support'. [Online]. Available: <https://support.bankid.com/sv>. [Accessed: 03-Jun-2016]
- [16] Symantec Corporation, 'Buy and Compare SSL Certificates'. [Online]. Available: <https://www.symantec.com/ssl-certificates/compare-ssl-prices.jsp>. [Accessed: 22-Apr-2016]

- [17] Microsoft, 'Preinstalled Trusted Root Certificates'. [Online]. Available: <https://technet.microsoft.com/en-us/library/cc962063.aspx>. [Accessed: 20-Apr-2016]
- [18] Hypersecu information systems, 'Hard vs. Soft Tokens: Making the Right Choice for Security'. Hypersecu information systems [Online]. Available: <https://www.hypersecu.com/downloads/files/whitepapers/HSTE-NB0012-RV1.0-Hard-Soft-Tokens.pdf>. [Accessed: 10-May-2016]
- [19] S. Santesson, W. Polk, P. Barzin, and M. Nystrom, 'Internet X.509 Public Key Infrastructure Qualified Certificates Profile', *Internet Request for Comments*, vol. RFC 3039 (Proposed Standard), Jan. 2001 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3039.txt>
- [20] R. Housley, W. Ford, W. Polk, and D. Solo, 'Internet X.509 Public Key Infrastructure Certificate and CRL Profile', *Internet Request for Comments*, vol. RFC 2459 (Proposed Standard), Jan. 1999 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc2459.txt>
- [21] T. Dierks and E. Rescorla, 'The Transport Layer Security (TLS) Protocol Version 1.2', *Internet Request for Comments*, vol. RFC 5246 (Proposed Standard), Aug. 2008 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5246.txt>
- [22] R. Housley, 'Cryptographic Message Syntax (CMS)', *Internet Request for Comments*, vol. RFC 5652 (INTERNET STANDARD), Sep. 2009 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5652.txt>
- [23] Microsoft, 'Understanding S/MIME', 16-Aug-2006. [Online]. Available: [https://technet.microsoft.com/en-us/library/aa995740\(v=exchg.65\).aspx](https://technet.microsoft.com/en-us/library/aa995740(v=exchg.65).aspx). [Accessed: 03-Jun-2016]
- [24] ETSI, 'Electronic Signatures and Infrastructures (ES); CMS Advanced Electronic Signatures (CADES)'. 650 Route des Lucioles, Sophia Antipolis France, Apr-2013 [Online]. Available: http://www.etsi.org/deliver/etsi_ts/101700_101799/101733/02.02.01_60/ts_101733v020201p.pdf. [Accessed: 07-Jun-2016]
- [25] Dawn M. Turner, 'eIDAS from directive to regulation', *Cryptomathic*, 15-Jan-2016. [Online]. Available: <http://www.cryptomathic.com/news-events/blog/eidas-from-directive-to-regulation-legal-aspects>. [Accessed: 27-Apr-2016]
- [26] Dawn M. Turner, 'Understanding the Major Terms Around Digital Signatures', *Cryptomathic*, 22-Jan-2016. [Online]. Available: <http://www.cryptomathic.com/news-events/blog/understanding-the-major-terms-around-digital-signatures>. [Accessed: 06-May-2016]
- [27] Sean Parkinson, Kathleen Moriarty, Michael Scott, Andreas Rusch, and Magnus Nystrom, 'PKCS #12: Personal Information Exchange Syntax v1.1'. [Online]. Available: <https://tools.ietf.org/html/rfc7292>. [Accessed: 10-May-2016]
- [28] J. Sermersheim, 'Lightweight Directory Access Protocol (LDAP): The Protocol', *Internet Request for Comments*, vol. RFC 4511 (Proposed Standard), Jun. 2006 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4511.txt>
- [29] ZyTrax, 'LDAP for Rocket Scientists', *Open Source Guide - LDAP for Rocket Scientists*. [Online]. Available: <http://www.zytrax.com/books/ldap/>. [Accessed: 17-May-2016]
- [30] ZyTrax, 'Chapter 3. LDAP Schemas, objectClasses and Attributes', *ObjectClasses*. [Online]. Available: <http://www.zytrax.com/books/ldap/ch3/#objectclasses>. [Accessed: 18-May-2016]
- [31] K. Zeilenga, 'Lightweight Directory Access Protocol (LDAP): Directory Information Models', *Internet Request for Comments*, vol. RFC 4512 (Proposed Standard), Jun. 2006 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4512.txt>
- [32] OpenLDAP, 'Introduction to OpenLDAP Directory Services'. [Online]. Available: <http://www.openldap.org/doc/admin24/intro.html>. [Accessed: 16-May-2016]
- [33] Adobe Systems Incorporated, *PDF Reference Adobe® Portable Document Format Version 1.7*, Sixth edition. Adobe Systems Incorporated, 2006 [Online]. Available:

- http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf
- [34] Adobe Systems Incorporated, 'Digital Signatures in a PDF'. Adobe Systems Incorporated [Online]. Available: https://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/Acrobat_DigitalSignatures_in_PDF.pdf. [Accessed: 02-May-2016]
- [35] Adobe Systems Incorporated, 'Digital Signatures Workflow Guide: a guide for workflow owners'. Adobe Systems Incorporated, 28-Sep-2012 [Online]. Available: http://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/Acrobat_DigSig_WorkflowGuide.pdf. [Accessed: 06-Jun-2016]
- [36] ISO, 'Document management: Portable document format'. ISO, 01-Jul-2008 [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502. [Accessed: 09-May-2016]
- [37] Stuart Haber and W. Scott Stornetta, 'How to Time-Stamp a Digital Document', in *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, London, UK, UK, 1991, pp. 437–455 [Online]. Available: <http://dl.acm.org/citation.cfm?id=646755.705358>
- [38] Riksdagsförvaltningen, 'Lag (2000:832) om kvalificerade elektroniska signaturer Svensk författningssamling 2000:2000:832 t.o.m. SFS 2011:803 - Riksdagen'. [Online]. Available: http://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/lag-2000832-om-kvalificerade-elektroniska_sfs-2000-832. [Accessed: 09-May-2016]
- [39] Andrew Betteridge, 'A Note on Electronic Signatures', *Ashfords LLP*, 15-Jan-2015. [Online]. Available: <http://www.ashfords.co.uk>. [Accessed: 24-May-2016]
- [40] Kommunikationsdepartementet, 'Digitala signaturer: en teknisk och juridisk översikt'. Kommunikationsdepartementet [Online]. Available: <http://www.regeringen.se/contentassets/4989aae001664fc6b89edde1b1918c71/digitala-signaturer---en-teknisk-och-juridisk-oversikt>. [Accessed: 09-May-2016]
- [41] Dan Puterbaugh, 'Understanding eIDAS – All you ever wanted to know about the new EU Electronic Signature Regulation | Legal IT Insider', 01-Mar-2016. [Online]. Available: <http://www.legaltechnology.com/latest-news/understanding-eidas-all-you-ever-wanted-to-know-about-the-new-eu-electronic-signature-directive/>. [Accessed: 24-May-2016]
- [42] EUR-Lex, 'Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC', 23-Jul-2014. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG. [Accessed: 24-May-2016]
- [43] C Adams, P Cain, D Pinkas, and R Zuccherato, 'Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)'. Aug-2001 [Online]. Available: <https://www.ietf.org/rfc/rfc3161.txt>. [Accessed: 07-Jun-2016]
- [44] John Ross, Nick Pope, and Denis Pinkas, 'Electronic Signature Formats for long term electronic signatures'. Sep-2001 [Online]. Available: <https://tools.ietf.org/html/rfc3126>. [Accessed: 07-Jun-2016]
- [45] Adobe Systems Incorporated, 'Protected View feature for PDFs (Windows)', *Protected mode*. [Online]. Available: <https://helpx.adobe.com/reader/using/protected-mode-windows.html>. [Accessed: 04-Jul-2016]
- [46] Adobe Systems Incorporated, 'Security (Digital Signatures)'. [Online]. Available: https://www.adobe.com/devnet-docs/acrobatetk/tools/PrefRef/Windows/Security.html?zoom_highlight=log#SignatureValidationLogging. [Accessed: 04-Jul-2016]

- [47] Jayakumar Thangavel, 'Digital Signature : Comparative study of its usage in developed and developing countries', Master's thesis, Uppsala, 2014 [Online]. Available: <http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A695339&dswid=9223>. [Accessed: 09-May-2016]
- [48] Johan Andersson, *Rekommendationer för Införande av Public Key Infrastructure*, Master's thesis. Linköping, Sweden: Linköpings Universitet, Avdelningen för Informationsteori, Institutionen för Systemteknik, 2002, LiTH-ISY-EX-3287–2002 [Online]. Available: <urn:nbn:se:liu:diva-1177>
- [49] Adobe Systems Incorporated, 'Philadelphia University helps students, faculty, and staff work more effectively' [Online]. Available: <http://www.adobe.com/content/dam/Adobe/en/customer-success/pdfs/philadelphia-university-case-study.pdf>. [Accessed: 07-Jun-2016]
- [50] Adobe Systems Incorporated, 'Pepperdine University, technology innovators in education' [Online]. Available: <https://www.adobe.com/content/dam/Adobe/en/customer-success/pdfs/pepperdine-case-study.pdf>. [Accessed: 07-Jun-2016]
- [51] Pace University, 'Pace University: Adobe Sign' [Online]. Available: <http://www.pace.edu/its/it-services-and-support/adobe-esign>. [Accessed: 07-Jun-2016]
- [52] The University of Arizona, 'Software Licensing for UA Faculty, Staff & Students' [Online]. Available: <http://softwarelicense.arizona.edu/adobe-sign-faqs>. [Accessed: 07-Jun-2016]
- [53] Adobe Systems Incorporated, 'University of Georgia enhances operations' [Online]. Available: <http://www.adobe.com/content/dam/Adobe/en/customer-success/pdfs/university-of-georgia-case-study.pdf>. [Accessed: 07-Jun-2016]
- [54] Blom Gunnar, Enger Jan, Englund Gunnar, Grandell Jan, and Holst Lars, *Sannolikhetsteori och statistikteori med tillämpningar*, 5:9. Lund: Studentlitteratur AB, 2011, ISBN: 978-91-44-02442-4.

Appendix A: Application form



BLANKETT

Blankettkod: UT-EXAR	Reviderad datum 2015-09-29	
Ansvarig avdelning UF:PLU		

ANSÖKAN OM EXAMENSARBETE/APPLICATION FOR DEGREE PROJECT

DEL 1/PART 1

Fylls i av studenten/*To be filled in by the student*

Förnamn/ <i>First name</i> XXXX	Datum/ <i>Date</i> 2016.04.22
Efternamn/ <i>Surname</i> YYYY	Personnummer/ <i>ID-number (YYMMDD-XXXX)</i>
E-postadress/ <i>E-mail</i> foo@kth.se	
Program vid KTH/ <i>Programme at KTH</i> Unknown	
Planerad start för examensarbete/ <i>Degree project is planned to start</i> 2016.04.22	

Fylls i av skolans administration/*Filled in by the school administration*

Studenten uppfyller poängkrav för att antas till examensarbete/*The student fulfills credit requirements for admission to degree project*

Datum/ <i>Date</i>
Underskrift behörig administrativ personal/ <i>Signature</i>
Namnförtydligande/ <i>Printed name</i>

Bifoga registerutdrag för studenten./*Attach transcript of records for the student.*



BLANKETT

Blankettkod: UT-EXAR	Reviderad datum 2015-09-29	
Ansvarig avdelning UF/PLU		

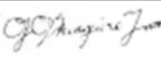
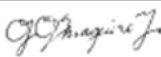
DEL 2, FÖRUTSÄTTER ATT DEL 1 HAR GODKÄNTS**PART 2, REQUIRES APPROVED PART 1****Fylls i av examinator/To be filled in by examiner**

Examinator/Examiner Gerald Q. Maguire Jr.
Kurskod* och kursnamn/Course code* and course name
Startdatum/Planned start

*Studenter antagna 2007-07-01 – 2015-06-30 kan välja graderat betyg. Ange då kurskod för graderat betyg./Students accepted for studies at KTH July 1 2007 – June 30 2015 may choose grading scale A-E. If this is the students choice, please use the associated course code.

Var ska examensarbetet utföras:/The degree project will be carried out:

På KTH, ange avdelning/At KTH, unit	Handledare/Supervisor
På företag, ange vilket/At a company, indicate name	Kontaktperson/Contact person
Utomlands, ange land/Outside Sweden, indic. country	Kontaktperson/Contact person

Signatur examinator/Signature examiner			Digitally signed by G. Q. Maguire Jr. Date: 2016.04.22 18:08:42 +0200
Signatur student/Signature student			

Förslaget godkänns som examensarbete/The proposal is approved as a degree project

Signatur grundutbildningsansvarig (GA)/Signature Director of First and Second Cycle Education
Namnförtydligande/Printed name

Ladok/Registration

Antagen till kurs, signatur	Namnförtydligande/Printed name
Registrerad på kurs, signatur	Namnförtydligande/Printed name

Appendix B: Sample form



Application form for thesis project

Part 1 (student):

First name: _____

Date: _____

Last name: _____

Signature: _____

Part 2 (educational administrator):

First name: _____

Date: _____

Last name: _____

Signature: _____

Part 3 (faculty member):

First name: _____

Date: _____

Last name: _____

Signature: _____

Appendix C: readme.txt

```
1 Hur man skapar ett digital ID
2
3 OBS! Innan du börjar ska du starta ett tidtagarur som mäter hur lång tid det tar för dig att:
4 * Skapa ett nytt digitalt ID och
5 * Läs och signera dokumentet
6
7 (Menyer och rubriker kan variera beroende på vilken version av Acrobat du har)
8
9 1) Starta Adobe Reader (exempelvis genom att öppna den bifogade .pdf filen "Application_form").
10 2) Tryck på "Tools" bredvid "Home" knappen, välj "Certificates" som har en grön ikon.
11 3) Tryck på "Digitally Sign" och dra ett fält med vänsterknappen för att skapa fältet du vill signera på (ta det fält som bäst överensstämmer för dig).
12 4) Nu öppnas ett fönster. Om du redan har skapat ett digital ID välj det du vill signera med. Om du inte har skapat ett digital ID tryck på "Add digital ID".
13 5) Ditt ID ska vara av PKCS#12.
14 6) Fyll i dina personuppgifter, algoritmen ska vara 2048 bitar och ID:t ska användas för både digital signering och data kryptering.
15 7) Nu väljer du vart ditt ID ska sparas samt vilket lösenord du vill ha.
16 8) Nu är ditt ID redo för signering, nu behöver du bara skriva in ditt lösenord varje gång du vill signera dokument med detta ID.
17
18 Skicka gärna ditt signerade dokument till mig och tiden det tog att skapa ett nytt digital ID samt tiden det tog att signera dokumentet.
19 Mvh Pontus
20
21 P.S. Din signatur innehåller endast din offentliga nyckel och personuppgifterna du angav i skapandet av ditt digitala ID. Din privata nyckel ligger sparad på
22 din dator och kommer inte att skickas med när du signerar dokument.
```


TRITA-ICT:EX-2016:148