NTNU

Norwegian University of
Science and Technology

Advanced Electronic Signature

Fazel Ahmad Azizi

Master of Science in Communication Technology
Submission date: June 2011
Supervisor: Stig Frode Mjølsnes, ITEM
Co-supervisor: Tord I Reistad, Difi

Norwegian University of Science and Technology
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

Advanced Electronic Signature

Master of Sicience in Communicaition Technology

Submission date:	June 11, 2011
Supervisor:	Stig Frode Mjølsnes
Co-Supervisor:	Tord I. Reistad

Norwegian University of Science and Technology
Department of Telematics

Advanced Electronic Signature

Fazel Ahmad Azizi

Norwegian University of Science and Technology
Department of Telematics

Problem Description

Advanced Electronic Signature

DiFi, Altinn and Lånekassen will implement a national digital signature to sign document submissions and mutual agreements. It is anticipated that a pilot will be launched in 2012.

A digital signature is very different to a hand signature, for instance how to establish what you actually sign. Moreover, the verification of a digital signature requires a correct and valid public key, whereas a handwritten signature is physically produced by a person.

The candidate of this project will try to understand the signature applications of Altinn and Lånekassen, then analyze the proposed digital signature architecture and standards to be used in the DiFi pilot and assess the utility and security of this solution compared to the existing Altinn "login signature".

Furthermore, the candidate will try to identify one or more parts of the architecture that can be given an alternative solution, and state the arguments that support that this will be an improvement. If time allows, experimental results in software that support the claims may be carried out.

Assignment given:	17.jan.2011
Supervisor:	Stig Frode Mjølsnes
Co-Supervisor:	Tord I. Reistad

Preface

This master thesis is carried out in the last i.e. 10th semester of the 5-year MSc in Communication Technology at the Department of Telematics, Norwegian University of Science and Technology (NTNU).

This work was performed in spring 2011 at NTNU in collaboration with Difi. The academic supervision was done by Professor Stig Frode Mjløsnes at Department of Telematics, with a co-supervision of Professor Tord Ingolf Reistad senior advisor at Difi.

I would like to thank Professor Stig Frode Mjølsnes and Tord I Reistad for their support and supervision during the work with the master project as well as for the valuable suggestions and comments.

I would also like to thank John Arild Amdahl Johansen, security manager at Buypass AS for his valuable information and spending time to answer some of my questions during the project work.

Fazel Ahmad Azizi

Abbreviations

AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CA	Certification Authority
CAdES	CMS Advanced Electronic Signature
CMS	Cryptographic Message Syntax
COBRA	Common Object Request Broker Architecture
CRL	Certificate Revocation List
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DSS	Document Security Store
ECP	Enhanced Client and Proxy
EEPROM	Electrically Erasable Programmable Read Only Memory
eID	electronic Identity
ELMER	Name of the Standard
ETSI	European Telecommunications Standards Institute
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol Secure
HTTPS	Secure HTTP
ICT	Information and communications technology
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
LDAP	Lightweight Directory Access Protocol
MD5	Message Digest 5
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OSI	Open Systems Interconnection
PDF	Portable Document Format
PADES-LTV	PDF Advanced Electronic Signature
PADES-LTV	PADES-Long Term Validation
PEPPOL	Pan-European Public Procurement Online
PGP	Pretty Good Privacy
PIN	Personal Identity Number
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
PS_1	Proxy Signer 1
PS_2	Proxy Signer 2

RAM	Random Access Memory
ROM	Read Only Memory
RP	Relying Party
RPC	Remote procedure call
RSA	Rivest, Shamir and Adleman
SAML	Security Assertion Markup Language
SEID	Samarbeidsprosjekt om eID og eSignatur
SEID-SDO	Signed Data Object
SHA1	SHA Secure Hash Algorithm 1
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SSO	Singe Sign ON
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TS	Technical Specification
TTP	Third Trusted Party
TU	Time Up
U_1	User 1
U_1	User 2
UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VRI	Validation Related Information
WAP	Wireless Access Point
WEP	Wired Equivalent Privacy
WSDL	Web Service Definition Language
XAdES	XML Advanced Electronic Signatures
XML	Extensible Markup Language

Figures

Tables

Definations

Electronic Signature:

The Norwegian law define electronic signature as any electronic data attached to other data which is utilized as authentication method [54].

Advanced Electronic Signature:

Advanced electronic signature is an extension of electronic signature where the following requirements must meet;

- Uniquely linked to the signatory5
- Has the ability to identify the signatory
- Created with the tools under the control of Signatory Is created using means that the signatory has control over, and is linked to other electronic data in such a way that subsequent changes can be detected [54]

Merchant (brukersted in Norwegian):

Merchant according to Mariam Webster dictionary has 3 meanings a buyer and a seller of commodities, the operator of retail business and one that is noted for a particular quality or activity [45]. The operator of a retail business applies in this thesis. For instance Altinn is merchant of Buypass, because Altinn is buying eID service infrastructure from Buypass.

Synchronous signing:

It means that the user is directed to Altinn signing page where the user will sign the document.

Asynchronous Signing:
In this context a user must manually login to Altinn where he must sign the document, or user must login to service provider's network and directed to Altinn for signing a document.

$\mathbf{E_{X_p}(h(m))}$: A a hash of message m is computed, where this hash is encrypted using private key, $\mathbf{X_p}$.

Thesis outline

The report is structure in 7 chapters. Chapter 1 presents an introduction to the thesis, Motivation for why this project is interesting, interpretation and scope of the research and what research method to be utilized.

Chapter 2 presents a brief review of related literature and theory to the project. It comprises PKI, Web Services, Electronic ID, the proxy signer concept which will be used in the solution proposal of this thesis. At the end of this chapter a brief review standards like SAML, SEID-SDO, PAdES-LTV, and SOAP.

Chapter 3 presents the existing signature solutions and Difi's solution proposals. It also presents the technologies to be used in the pilot project, where it further considers the analysis of these technologies to discuss the weaknesses.

Chapter 4 presents solution proposals of this thesis, where 3 solutions, based on Proxy Signer concept, are proposed. Meanwhile advantages and disadvantages of all solutions are discussed. It further considers the technologies to be utilized, if it is implemented, to discuss how to cover the shortcoming and security weaknesses, compared to Difi's solution proposal.

Chapter 5 Analyze the solution proposals of this thesis. Some requirements are developed in order to discuss that the system satisfies all functionalities signature system requires. The requirements are analyzed by Use Case diagrams.

Chapter 6 validates all functionality of the system against the requirements, and it further validates the model against the principals set by Difi.

Chapter 7 presents conclusion of project work and discusses remaining work for future.

Contents

Chapter 1

Introduction

This chapter introduces the research problem of this thesis. It will also answer how the assignment of this thesis is interpreted, as well as what research method is intended to be used. At the end of this chapter a brief outline of the report is given. In this report the words eSignature, electronic signature is referred to as advanced electronic signature.

1.1 Motivation

Signature in our society is indication for acceptance of a condition. In our daily life we sign documents for different purposes. A primary function of a manuscript signature (signing a physical object) as described in [62] is to provide the identity of the signatory, describe the intention of the signature and accepting the content of the document. The secondary functions are to validate official action and protecting both the signatory (consumer) and service provider (the one that demands a signature).

When two people make a deal and they agree to sign a document in order to accomplish the deal and to cover that neither party can deny the deal later. What if one party is using a fake signature and making a plan to deceive the other party? There is a legal solution for this kind of deal, using either a third party as a witness or using the Norwegian concept of public notary (publicus notaries) in order to cover non-denial action by both parties.

The world is on its way to becoming more digital. Things and processes done in the real world will more and more be done electronically. This is the case with signing which will go from signing a written agreement by hand to signing electronically. What is an electronic signature? Electronic signature is an output of a mathematical function where the document or part of it is given as an input. Securing an electronic signature is done through cryptographic means. Paying by card, trans-

action using internet banking, singing in the screen for a postman when one gets recommended delivery and so on are actually signing digitally/electronically.

Signing electronically is very different to signing by hand. The treats are different, there establishing that what you see is what you sign is problematic, and there are many other aspects that has to be examined when signing is done in a national or international setting.

1.2 Thesis Interpretation and Scope

As the problem description states that Difi, Altinn and Lånekassen will be implementing a national electronic signature system to sign document without being present.

First of all what is electronic signature? As described in [54] electronic signature is data in electronic form attached with other electronic data and utilized as an authentication method. As described more over that there is an existing method for signing document then why do we need new mechanism? The main reason is that the existing mechanism as defined is not adequate, and the demand is more than what is described over. Of course there is need for new signature to cover all needs which is called advanced electronic signature.

Advanced electronic signature is an extension of electronic signature with extra requirements. Is there any need for developing advanced electronic signature?

A working group at Difi is given the job to investigate whether there is need for an advanced electronic signature. This group concluded that there is a need for it, and therefore they proposed new solutions [20].

In this thesis it is necessary to understand how the existing signature application works, and analyze the Difi's proposed solutions. In order to analyze it effectively, it is necessary to understand what technologies and standards is to be used and whether there is any weaknesses in those technologies. How to cover those weaknesses would be included in proposed solutions of this thesis.

The thesis's assignment can be split into two parts;

- Understanding Altinn and Lånekassen existing signature application and analyzing the Difi's proposed solutions
- Propose an alternative solution which could be an improvement to Difi's solutions

1.3 Overview of the Result

In this project I will try to analyze Difi's solution proposal in order to assess the utility and security of their proposal in compared to Altinn "login signature". For analyzing I must study the technologies to be used in their pilot project. I will find the security weaknesses and other shortcoming related to the technologies and standard to be utilized.

The aim of this project will be to design a new system, based on PKI technology, to cover the shortcoming found in analysis. To come up with a new I idea to cover the found shortcoming, a broad study of relevant technologies must be carried out. Since Difi also set requirements and principals to be follow. In order to take this in consideration I must study all possible technologies and come up with on to for designing the new model where it could satisfy the principals set by Difi.

1.4 Research method

Academic research reports has four potential properties where one of them is that the research must be supported by potential research papers which are applicable and based on logical theory [46].

Three questions are presented by [18] to be answered in by a research proposal:

- What? What is the research's purpose i.e. what is the researcher trying to find out?
- Why? Why is this research so important i.e. what is the researcher's motivation?
- How? How will the researcher answer the research questions?

In this research a combination of qualitative and quantitative research method is utilized. Qualitative research is all about exploring issues, understanding phenomena, and answering questions. Quantitative research method is to investigate the quantitative properties of system and its relationship with other entities. The main objective of this method is to develop and employ mathematical models. For conducting the research the approaches utilized are presented in section 1.4.2.

1.4.1 Reviewing Relevant Literature

For reviewing literature it is needed to collect potential literature in order to support the claim of the proposal. For collecting literature BIBSYS, Google scholar and IEEE magazine in order to collect potential papers and text books relevant to the theme of the thesis are used.

It is necessary to evaluate the papers' quality, because low quality papers will weaken the research.

1.4.2 Reviewing & Analysis of Preliminary research

Reviewing and analysis of preliminary research aims to understand how the system works and how the researcher stated the argument for his research conduction.

Technologies and standards are analyzed in order to both understand them and come up with a new idea for developing an alternative solution as an improvement.

1.4.3 Developing new Model

Developing is the most critical and difficult part of research where the researcher must prove his claim by comparing the result with the known technologies and standards.

Difi is launching a pilot project by 2012 with the technologies and standards to be utilized against requirements specifications set by Difi. In this research the result of development is compared with technologies, standards and Difi's requirements specifications in order to claim the proof of the research.

Chapter 2

Theory

In order to come up with solution to improve Difi's proposed solution, a broad study of the following technologies and standards are carried out; PKI, web services, electronic identity (eID), SAML, SEID-SDO, PAdES-LTV and SOAP.

The named technologies and standards are used in the existing solution except PAdES-LTV standard. This standard will be used in the Difi's pilot project instead of SEID-SDO standard. These technologies are intended to be used by the proposed solutions of this thesis also except SEID-SDO which would be replaced by PAdES-LTV.

Most of the technologies and standards are used in common where SEID-SDO is excluded in the new solutions of Difi and this thesis where PAdES-LTV is utilized instead.

Therefore all these technologies and standard are briefly described in this chapter.

2.1 Public Key Infrastructure (PKI)

Public key infrastructure is framework consisting of policies defining the rules under which the cryptographic systems operate and procedures for generating and publishing public key certificate. Public key infrastructure consist certification and validations operations. Certification connects a public key to a person while validation guarantees the validity of the certificate [76].

This section describes the principles of public key cryptography.

2.1.1 Public Key Cryptography

The concept of public key cryptography is used as long as humans are expecting secure communications with each others. Many techniques are developed over

thousands of years for encrypting message in order to secure the confidentiality of the message over the communication link from potential eavesdroppers [15].

Each technique used pre shared secret keys for encrypting and decrypting the message. Encrypting message (plain text) produces cipher text (non readable text) and decrypting cipher text produces the message (plain text). This technique is called symmetric encryption.

In the beginning of 1970 a new encryption technique was developed by Whitfield Diffie and Martin Hellman where two different keys were used for encrypting and decrypting [78]. The encryption key is called private key and the decryption key is called public key and both of them are called private-public key pair, and the technique is called asymmetric encryption. Private Key is the secret while public key, as the name reveal, is public i.e. it is published. The invention of this technique opened a new chapter in the cryptography world.

Public key cryptography is a powerful tool that allows for authentication key distribution and non-repudiation [76]. A fundamental problem raises here that how to believe that the public key belongs to the one who claims for, especially when one is accessing public key.

A public key cryptography is useful in commercial applications when the public key is tracked efficiently. This is accomplished through the so called public key infrastructure (PKI).

A brief description of some entities in PKI is given below;

Certificate

A traditional definition of certificate according to [8] is that it is a document testifying the truth of something. The owner is identified through picture, handwritten signature and other physical cues in certificate.

In the same way as in a paper-based certificate, in digital world, certificate is an electronic document or digital stamp testifying the identity of the holder of certificate. In other words a certificate is digital document binding an entity name to a public key [76]. A certificate is issued by certification authority. A certificate contains all necessary security parameters for identifying the holder and proving the originality of the certificate. The issuer of the certificate guarantees the authenticity of the holder of certificate. As an example SSL/TLS certificate will be discussed.

SSL/TLS Certificate

SSL is originated by Netscape. The main idea behind this protocol was to establish secure channel between two entities. It has several versions where the last version according to [77] is V3.1. This protocol utilized TCP to provide reliable and secure

end-to-end secure services. It is a two layers protocol which operates on the top of TCP. Figure 2.1 illustrates the protocol stack.

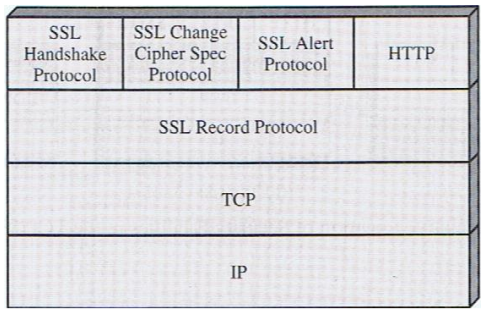

Figure 2.1: SSL Protocol Stack [77]

The important concepts of SSL are connection and session. Providing service between two entities according to OSI definition is transport which in this case is peer-to-peer [76]. The session is association between client and server which is created by hand shake protocol, and is secured through cryptographic means and can be established between different entities [76]. Figure 2.2 illustrates how SSL works and a deep detail is omitted here.

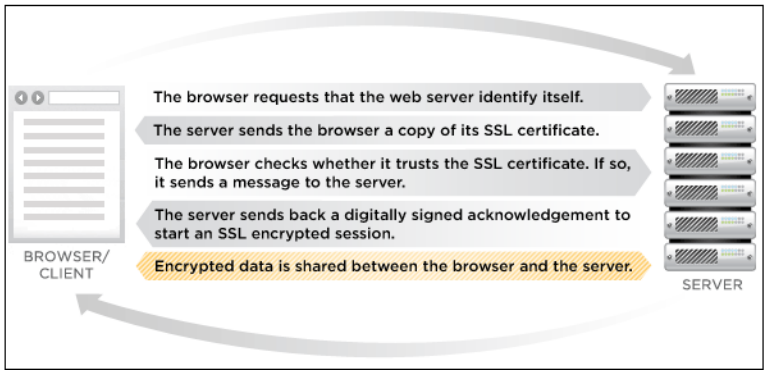

Figure 2.2: SSL Connection set up [75]

For securing the content (confidentiality) symmetric encryption and key exchanging asymmetric encryption are used. For connection's reliability message includes integrity check i.e. hash of message is computed and signed.

TLS is predecessor of SSL. IETF established a working group to standardize SSL to be adopted in the internet environment. There are slightly technical differences between SSL and TLS which is not discussed further here.

Certification Authority (CA)

To ensure that the communication is secure using public key cryptography a third trusted party is needed to attest the authenticity of the communication parties. This is achieved through the concept of certification authority (CA) where CA certifies that public keys to be used for decrypting belong to the claimant.

A physical entity who issues such certificates which identifies the owner of a public key is called certification authority. In other words a CA certifies that a public key belongs to physical entity. This is called identity binding or public-key certificate.

Relying Parties (RP)

Relying parties are entities that are trying to identify the identity of another party using digital signature. In the real world a relying party would be a service provider who wants to identify its customer in order to provide service to the right person.

Certificate Revocation Methods

Certificate revocation is a process for changing the status of a certificate issued by CA.

An issued certificate is expected to be in use for a period of time for which it is valid for, but in some circumstances certificate validity may expire due to different reasons for example an employee is no longer working for an employer where the employer is needed to cancel his certificate [63]. This is achieved through two different protocols, Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP).

In CRL protocol each CA issues a data structure periodically, called Certificate Revocation List, and made it publicly available through repository for the end users. The list identifying the revoked certificates is time stamped [63].

When a timely revocation of certificate is necessary OCSP is a better solution. OCSP enables the applications to check the status of a certificate from an OCSP responder. In some cases where the information is sensitive the revocation information of a certificate must be clear before a communication can take place. OCSP gives timely revocation information of certificate than achieved through CRL [47]. These two methods contribute in finding the status of a certificate whether it is valid or revoked.

2.1.2 Public Key Certificates

A public key used by many users require confidence that the private key associated with a particular public key belongs to the one who claims for. This is achieved

through public key certificate [63].

A public key certificate is a document issued and signed digitally by CA. The intentions of having a certificate are identity binding and ensuring the integrity of a public key in distributed PKI environments.

Different certificates have different formats. The first public key certificate X.509 is proposed by IETF in 1988 and intended to be used by the internet community where the newest version of the named certificate was last updated in april 2002.

Every public key certificate has finite lifetime [15], then the key becomes revoked automatically. PKI must maintain continuous update of the system.

Revoked keys are stored by the user due to needs of decrypting the document encrypted with the private key connected to revoked public key.

2.1.3 Proxy Key Certificate

Proxy certificate is certificate derived from, and signed by a normal X.509 Public Key certificate or by another proxy certificate [67]. In this case the original signer certifies proxy signer's public key certificate. In this way the delegation given by original signer will be trusted by the receiver, because the original signer has a normal and valid public key certificate. This is achieved through signing proxy certificate by the original signer's certificate. Original signer's certificate is certified by a known CA. The approach for deriving public and private key pair is described in section 2.4.1. When the proxy signer is going to sign on behalf of original signer, it will use the proxy certificate for authentication of proxy signer's public key, because in this certificate the validation path will reach the CA which certified the original signer's certificate.

2.2 Web Services

Web services are an important part in web technology describing computational functionality that can be found and invoked over any network [22].

One of the important feature of web services is that web service is self-describing i.e. is readable by both human and machine, and self-contained i.e. constituting a complete and independent unit in and of itself, modular application that can be published, located and invoked over network from anywhere in the world using world wide web.

Web services are designed to be used by other application or program rather than humans. It can be accessed via internet protocols like http, ftp etc. Different web

services can be mixed and matched with each other in a value chain.

The main usage areas of web services are;

- Application integration web services
- Business integration of web services
- Commercial web services

According to [22] Web services framework is divided into four areas where specifications are developed in each area;

- *Publication* of service i.e. services to be published are made easily available through making mechanisms to find, for example Universal Description Discovery and Integration UDDI standard can be used to find a published service.
- *Discovery* of services where a service with a certain functionality is made searchable *Description* of services i.e. a full description of a service for the user in order to give the User the notion of how to invoke the service, for example Web Service Description Language WSDL standard is widely used for the description of services.
- *Invocation/communication* of services where a protocol that allows invocation of web services in the internet and communications among web services, for example SOAP Simple Object Access Protocol standard is used for these purposes.

2.3 Electronic ID (eID)

Electronic ID (eID) is a type of smart card which is considered to be one of the most important technologies for the modern information system. Smart card technology has widespread utilization in different information system such as MasterCards, Visa cards and eID cards. The main reason for wide utilization of smart cards in authentication process is the security services provided by it.

There are two main types of smart cards, memory-based chip card and microprocessor-based chip card.

Memory-based chip card

The application where the functions of the card is fixed uses this kind smart card[12] and information can be stored once in the card [41]. The card uses synchronous protocol to communicate with a card reader [12]. Almost no security gains are

discovered so far compared to magnetic stripe cards. On the other hand magnetic stripe cards can easily read the stored value in the card and make a copy of it [41], [7].

However the arithmetic logic unit of the card is very small but is able to perform necessary cryptographic operation for authentication and encryption [79].

Micro-based Processor Chip Card

This type contains a microprocessor and different types of memory, Input/output (I/O) circuits and an operating system in order to handle all operations as a mini-computer. For enhancement of crypto operations speed a small crypto microprocessor can also be added.

Different types of memory given in [41] are as follows;

- Read Only Memory (ROM) as the name reveals, contains data to be only read and stored during the production and is 16KB
- Random Access Memory (RAM) contains volatile type of data which could be erased as a result of shut down or overwriting, and is 512Bytes
- Electrical Erasable Programmable Read Only Memory (EEPROM) contains non-volatile data which could be both erased and programmed by an application, no data is lost in shut down process

These kinds of cards are used in the electronic Identity (eID) cards.

2.4 The Proxy Signer Concept

The concept of proxy signature is described in more detail because in this thesis the main focus on alternative solutions to signature schemes are based on this concept. In order to have a good understanding of the concept it must be described in more detail.

The concept of proxy signature is a signature scheme where an original signer delegates his signing capability to another signer called proxy signer. This concept is introduced and studied first by Mambo et. al. in late 20th century [49]. This scheme has been evolved since then. Proxy signatures' requirements must be present in order to trust this scheme and are defined according to [39], [42] and [44] as follows;

Strong unforgeability; a valid signature can be created by proxy signer on behalf of original signer which cannot be recreated by the original signer and neither by a third party undesignated as a proxy signer.

Strong Verifiability; From a signature created by proxy signer the verifier can be convinced that there is an agreement of delegating signing capability to a proxy signer on a signed message.

Strong identifiability; Identity of a proxy signer and the original signer can be determined from proxy signature.

Strong non-deniability; when a valid proxy signature is created on behalf of an original signer the proxy signer cannot deny that the signature is not created by the proxy signer.

Prevention of misuse; A proxy signing key (private key) cannot be used for other purposes than it was created for.

This scheme is useful for devices lacking necessary computational power for computing heavy cryptographic computation in real time where these devices can use a server as proxy signer to perform such computation on behalf of him [2].

Mambo's et. al. scheme do not satisfy the first property of this concept. In Mambo's scheme there is no authentic information in proxy signer's key pair. This is a weakness of the scheme which divides this scheme in to strong and weak signature. In [42] they further classify this scheme into designated, non-designated and self-proxy signature where in designated scheme an original signer specify a proxy signer in a proxy key issuing stage by adding the proxy signer's ID in original signer's signature parameters. Non-designated and self-proxy schemes are not desirable for this thesis.

According to delegation signing capability proxy signature is classified in to the following;
Full delegation where original signer and proxy signer share the same private key i.e. original signer gives his private key to his proxy signer for signing on behalf of him.

Partial delegation where original signer derives private key to proxy signer in order for the proxy signer to use it to sign on behalf of original signer, but proxy signer cannot derive original signer's private key from it. This has an advantage, if a proxy signer's private key is compromised but the original signer key is not compromised.

Delegations by warrant where the original signer make a warrant composed of part of the document and part of public key and send it to proxy signer. Proxy signer uses the corresponding private key to sign on behalf of the original signer. The weaknesses in previous types can be overcome by using warrant [66].

In [2] delegation by certificate is mentioned where the keys are certified by certifi-

cate to make them trustful. It means that the original signer will be CA for the proxy signer and will be in the path of certificate.

2.4.1 Key Issuing

The key issuing depends on what kind of delegation is desirable. In this section the key issuing scheme desirable by this thesis is described in detail.

In full delegation then the original signer's private key is shared with the proxy signer but this has some disadvantages if key is compromised by proxy signer then the original signer's key is also compromised [66].

In a partial delegation original signer derive a private key from his own private key such that proxy signer cannot derive original signer's key from it. An advantage of this scheme is computation speed. A disadvantage of this scheme is that a proxy signer can forge the key i.e. proxy signer can misuse and sign what he wants and pretend that it is carried out on behalf of original signer.

In partial delegation by warrant the original signer derive a key and make a warrant from part of the message and its public key and sends it proxy signer for signing on behalf of him. This scheme improves the disadvantages of previous schemes, but it also has a disadvantage of extra computation.

Before going more into detail of partial delegation with warrant some cryptographic parameters must be described which will be used throughout this and next sections.

$\mathbf{p}$ is a large prime and $\mathbf{g}$ is generator of multiplicative subgroup of $\mathbf{Z^*_{p-1}}$. Both $\mathbf{p}$ and $\mathbf{g}$ are public while $\mathbf{X_p}$ is private. $\mathbf{h(m)}$ is a hash function computing hash of message $\mathbf{m}$. $\mathbf{X_u}$ and $\mathbf{Y_u = g^{X_u} \bmod p\text{-}1}$ are original signer's (User1 in this thesis) private and public pairs respectively. $\mathbf{m_w}$ is warrant message where the identity of original signer and its public key and information on delegation are given in a signed form ($\mathbf{m_w}$ in this thesis is different and is described later).

The key issuing in partial delegation is as follows;

The original signer chooses a value $\mathbf{k} \in_R \mathbf{Z^*_{p-1}} \backslash \{0\}$ and computes $\mathbf{K = g^k \bmod p}$. The original signer computes hash of $\mathbf{K}$ concatenated with $\mathbf{m_w}$, i.e. $\mathbf{e = h(K||m_w)}$. The original signer further computes $\mathbf{X_p = e.X_u + k \bmod p\text{-}1}$.

Original signer sends ($\mathbf{X_p}$, $\mathbf{K}$, $\mathbf{m_w}$) to Proxy Signer in a secure manner where $\mathbf{X_p}$ is the proxy signer's private key to be used for singing on behalf of original signer. Along with these tuples $\mathbf{Y_u}$ (a certified public key of original signer), $\mathbf{g}$ and $\mathbf{p}$ are sent to proxy signer. Public key of proxy signer is derived by verifier as follows; $\mathbf{Y_p = Y_u^e . K \bmod p}$. Proxy signer checks for the validity of private key $\mathbf{X_p}$ as

follows;

Compute hash of $\mathbf{K}$ and $\mathbf{m_w}$ i.e. $\mathbf{e{=}h(K||m_w)}$.

Accepts the key if $\mathbf{g^{X_p} =^? Y_u{}^e .\ K\ mod\ p}$, otherwise reject.

Private and public keys for both original and proxy signers are derived as shown above. Public key of original signer is certified by CA where public key of Proxy signer is derived i.e. certified by original signer , this is called proxy certificate [rfc3829]. A path hierarchical trust will be modeled.

2.4.2 Signing the Message and Verifying by Counterpart

For signing a document the proxy signer can compute the hash of the document m and sign it as follows;

$$\mathbf{Sig = E_{X_p}(h(m))}$$

The signature carried out by the proxy signer is ($\mathbf{m}$, $\mathbf{Sig}$, $\mathbf{K}$, $\mathbf{m_w}$)), and is sent to counterpart.

The counterpart verifies the signature in the same way as verification of original signature scheme with some extra computation as follows;

Compute

$$\mathbf{e = h(K||m_w)}$$

Derive public key of proxy signer;

$$Y_\mathbf{p} = \mathbf{Y_u^e .\ K\ mod\ p}$$

Accept if;

$$\mathbf{D_{Y_p}(Sig) =^? e}$$

otherwise reject.
This scheme is called non-protected partial delegation signature scheme where both the original and proxy signer can create a valid signature.

2.5 Standards

In this section the standards used in Altinn existing solution, to be used in Difi's proposed solution, and to be used in proposed solutions by this thesis are described. Brief overviews with a deep description of the part discussed in analysis are given.

2.5.1 Security Assertion Markup Language (SAML)

SAML is an XML based open message standard that encodes security assertion and corresponding protocol messages in of course XML format. In other words SAML is an XML based framework for exchanging security information online between partners. This can further be précised that it is XML framework for creating, requesting and exchanging security assertions between the entities [58]. The message standard is defined in [57].

In SAML protocol binding that embeds SAML construct in other structure for transport is allowed. Building on the SOAP with its SOAP over HTTP binding method is an example, where SOAP over HTTP binding is a HTTP request/response method that complies with SOAP encoding rules [82]. This standard includes description of the use of SAML assertion in the communication protocol and framework.

SAML Architecture

SAML consists of building blocks which all together support a number of scenarios. These components support transport of authentication, authorization and identity information between autonomous entities. A brief review of basic components of the SMAL is given in index A.

Single Sign-On (SSO) Open Standard

SSO standard is developed by the Organization for the Advancement of Structured Information Standards OASIS in 2002 [59]. The intention of developing of this standard was to give Users the opportunity of avoiding sign on several times while using the same authentication unit. It is defined in different ways with the same result that one authentication is adequate for accessing several applications simultaneously. The definitions are follows;

- A specialized form of software authentication that enables the User to authenticate a single time to get access to a variety of other applications.
- SSO is a session/User authentication process that gives the User the permission by giving one username and password, to authenticate for gaining access to multiple applications.

The technologies that can use to provide a single sign-on can be divided into three main categories [35] that are ticket-based, cookie-based, and PKI-based where every mechanism has both its advantages and disadvantages. PKI-based and ticket-based are described here.

Scripting is a cookie-based mechanism to implement SSO and is as follows;

A client-side script in the user's workstation mimics the procedure for logging on to the application or application server, sending the username and password across

the network when the user has given them. This mechanism in addition to mimic the user's input for logging on, is also able to understand and responds to messages from application server for example if the password change is requested it will update changing of password with all application sharing the authentication data. In the case of error this script would need to interpret the error and send the correct response. The greatest advantage is that it gives a short term solution. A possibly fatal disadvantage of script is that rather than increasing the security it may decrease the security if the user is unlucky for example if the password needs to be stored for any length of time on the workstation. Another problem is that the script is extremely sensitive to response returned by the target system for instance changing Login to login can cause a script failure [31]. Eavesdrop attack was likely to be carried out, but this problem is overcome through secure authentication method where challenge and response mechanism are used.

Another cookie-based method for implementing SSO is using OASIS SSO standard. SAML's SSO Browser Artifact profile is utilized for achieving implementation of SSO. SSO Browser Artifact profile describes the usage of SAML message to perform SSO operation. A profile consists of SMAL protocol and bindings. A second profile called Attribute profile includes for example LDAP, DCE information carried with assertions [58]. Two other attribute are used in building a system metadata where sharing of configuration information between two communication entities is defied with and authentication context where retrieving extra information needed by the service provider is carried out. Figure 2.3 below shows how the SSO operation is carried out.

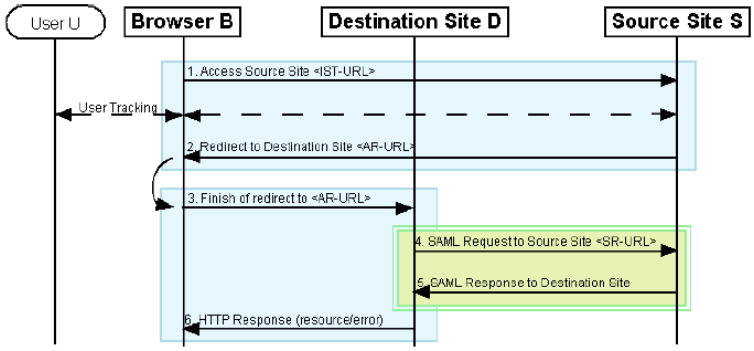

Figure 2.3: Work flow in Single Sign on [73]

When a user is authenticated via an eID provider then the site that requested the authentication will store an assertion about the user's identity for using it later. When the user wants to browse to another site where authentication is mandatory then this site includes the so-called SAML artifact profile data into the redirect that refers to the assertion stored. When the new site gets the redirect with the

artifact data it will request the previous site for sending the corresponding assertion from it. By sending the asserted data from the previous site to the new site, the previous site confirms the identity of the user [56].

Single Logout (SLO)

Single Logout is a reverse operation of Single Sign On where the user request the eID provider to logout, eID provider will decouple all sessions associated with this user's username and password. A deep description is omitted here.

2.5.2 SEID-SDO

Signed data object SDO developed first by Bank industry in Norway called BankID-SDO where it is further developed by samarbeidsprosjekt om eID og eSignering (SEID). It is thereafter called SEID-SDO. SEID-SDO is further development of ETSI standards ETSI TS101 733(CAdES) and ETSI TS 101 903(XAdES). Since ETSI standard are XML based data structure then SEID-SDO is also XML based standard which supports three types signature structure to be saved [68].

SEID-SDO is a XML based data object and semantic used for saving information where this information must be understandable and useable by an arbitrary part. The data object must be able to keep the standard-based electronic signature and its corresponding validating data as a base for long-term storage and validating the signature over a long period.

A document of a signer could be of any type data object for example a PDF document, an ASCII string, a XML format SEID-SDO object and so on. In order to provide increased security value with respect to signature validation in retrospect rather than computing a hash of just the document, hash of some validation's attributes are also included. An example of attribute could be reference to the signer, signature certificate, signature policy etc. [68]. Figure 2.4 shows the SEID-SDO object structure.

SEID-SDO is preferred oven for ETSI's standards and is recommended. The reason for this could be many factors where some important are included here;

As an extension to ETSI standards SEID-SDO offers a unified XML structure and thus a common exchange format that encompasses both ETSI compliant signature and BankID-SDO signature object [68]. Using SEID-SDO will have another advantage that different types of signature are supported.

A detail description of the structure is omitted here but a brief list of signature elements of profiling are included is in table in figure 2.4. The table is fetched from [68].

Profile name	Profile Description	Analog to ETSI profile	Purpose of Profile
SEID-SDO - Basic	Minimum profile with minimum signature and minimum sett of mandatory data	ES(BES/EPES)	Storing of Basic Signature without Validation data
SEID-SDO-Basic-V	SEID-SDO –Basic +additional requireme-nts to certificate valida-tion data	Not covered by ETSI	Storing of Basic Signature without Validation data. Introduced due to covering BandID's needs
SEID-SDO-Basic-T	SEID-SDO Basic + requirements about timestamp signature value	ES-T	(not shown here)
SEID-SDO-TSP	SEID-SDO Basic + requirements about Data validation certificate	Not covered by ETSI	An alternative use of SEID-SDO-Extended if one utilizes signature validation as a 3rd party trust service
SEID-SDO-Extended	SEID-SDO Basic + requirements about complete validation data connected to corresponding Basic Signature	ES-X-Long	(not shown here)
SEID-SDO-Archive	SEID-SDO Basic + requirements about Timestamp over both signature value and complete validation data	ES-A	(not shown here)

Figure 2.4: SEID-SDO profiles [68]

The outer layer of SEID-SDO shows that inside the SDO signature object three types of signature elements are taken thus CMS signature element, CAdES signature element and XAdES signature element. This means that this object supports three types of signatures based on three different signature standards. CMS is IETF syntax where CAdES and XAdES are ETSI's standards. Figure 2.5 shows the outer layer of the SEID-SDO structure.

Timestamp and its value with the TS certificate and revocation data are packed in an object called SDOTimeStamp.

SDO signature object and SDOTimeStamp are put together in a single object called SDO Seal.

SEID-SDO version, SDO data part, SDO Seal, Metadata and signed object are embedded inside the SDO object. At the end many SDO objects can eventually be saved in a single object called SDO list.

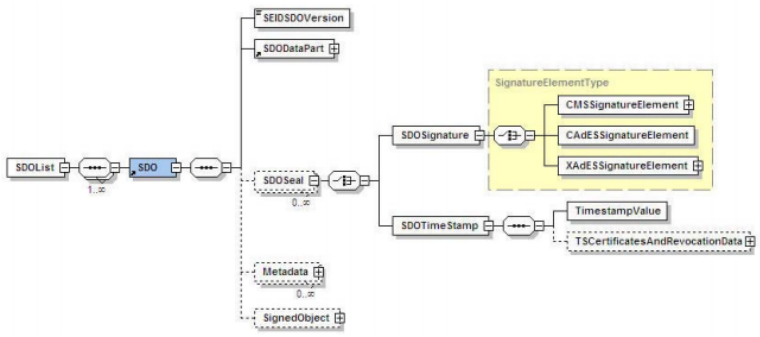

Figure 2.5: SEID-SDO object [68]

2.5.3 PAdES-Long Term Validation

In order to validate an electronic signature the validation data must be associated with the signature such as certificate, Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP). The latter validation data OCSP is an online service used to check the status of a certificate.

If a certificate expired after it had been used for the verification of a signature (with long term verifiability), the signature is no longer verifiable.

PDF Advanced Electronic Signature (PAdES) is a signature format standard developed by ETSI and widely used for including the signature on PDF document [26]. It is an extension of PDF where additional features are added.

PDF Advanced Electronic Signature - Long Term Validation is an extension of PAdES where long term validation features included. It is widely used as saving and transferring format for electronic documents where validation of signature in long term is desirable.

PAdES- LTV uses an extension to ISO 32000-1 [36] called Document Security Store (DSS) to carry the validation data of a specific signature for validating the signature in retrospect, with an optionally Validation Related Information (VRI) which relates validation data to a specific signature [24]. The figure 2.6 below illustrates the DSS and VRI structure.

In order to save the time when the signature is carried out the document must have timestamp where it will also show when the document was created first. PAdES - LTV uses another ISO 2000 -1 extension called Document Time Stamp to extend the lifetime of document for protection [25].

The DSS is first appended to the document where the timestamp is of the whole document, i.e. both PDF document and DSS, and then the timestamp is added as

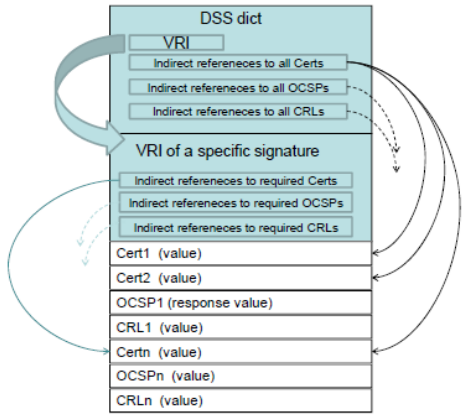

Figure 2.6: DSS and VRI Structure [25]

a document timestamp to the last provided document as illustrated in figure 2.7 below.

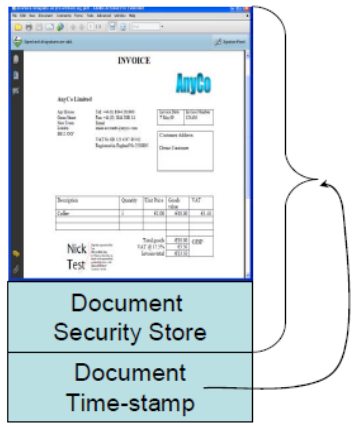

Figure 2.7: DSS and Timestamp Document added [25]

The validation data could be further protected by making a new DSS where the timestamp document is embedded in the DSS and then taking the timestamp again and appending to the document as illustrated in figure 2.8 below.
A deep description of PAdES - LTV is given in [25] part 4.

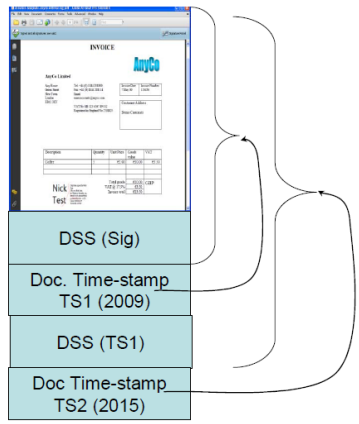

Figure 2.8: Further Protection of Validation data [25]

2.5.4 Simple Object Access Protocol (SOAP)

Simple Object Access Protocol SOAP is communication protocol which communicates two applications [81] and is W3C recommendation.

Simple Object Access Protocol SOAP is a protocol used for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies for defining extensible messaging framework, and providing a message construct that can be exchanged over many different underlying protocols. Advantages of using SOAP are simple, extensible, and it is independent of any peculiarly programming model and other implementation specific semantics.

SOAP is one way asynchronous communication technology which adapts several types of message-passing mechanism such as RPC, document oriented, publish and subscribe and so on [28].

A deep description of SOAP can be found in for instance [28], [82]

Chapter 3

Signature Solutions

There exists a mechanism for signing documents online in public sector in Norway. Altinn offers the so called "login signature" mechanism for service providers. Since Altinn's existing solution does not fulfill the requirements of an advanced electronic signature then Difi was given the job to develop a signature scheme on national level. In this chapter Altinn existing signature solution, Difi's proposed solutions are described. Furthermore Difi's solution is analyzed.

3.1 Existing signature mechanism of Altinn and Lånekassen

This section presents the so called "login signature" existing solution maintained by Altinn and utilized by both Lånekassen and other public sectors in Norway.

Altinn and lånekassen use the authentication based signing (login signature) mechanism. The figure 3.1 shows a high level description of signing mechanism implemented by Altinn.

This signing solution is based on a form sent to Altinn from end user system. Today BuyPass eID is used for signing but a technical integration for use of BankID is also implemented.

In this solution data is sent using web-services from user system in XML format according to form specification. user with signing rights must log on to Altinn and gets the form from his inbox and goes to the signing step. The user chooses to either see the data in the document and then sign or just sign without reading.

When the user chooses to sign, an applet from BuyPass will appear to sign and give a pin code in applet. Signing is carried out in an applet in an interaction with Buypass. Then the form is sent back to user system and a copy is saved in Altinn's archive. Deniable information is saved in Third Trusted Party (TTP) [20].

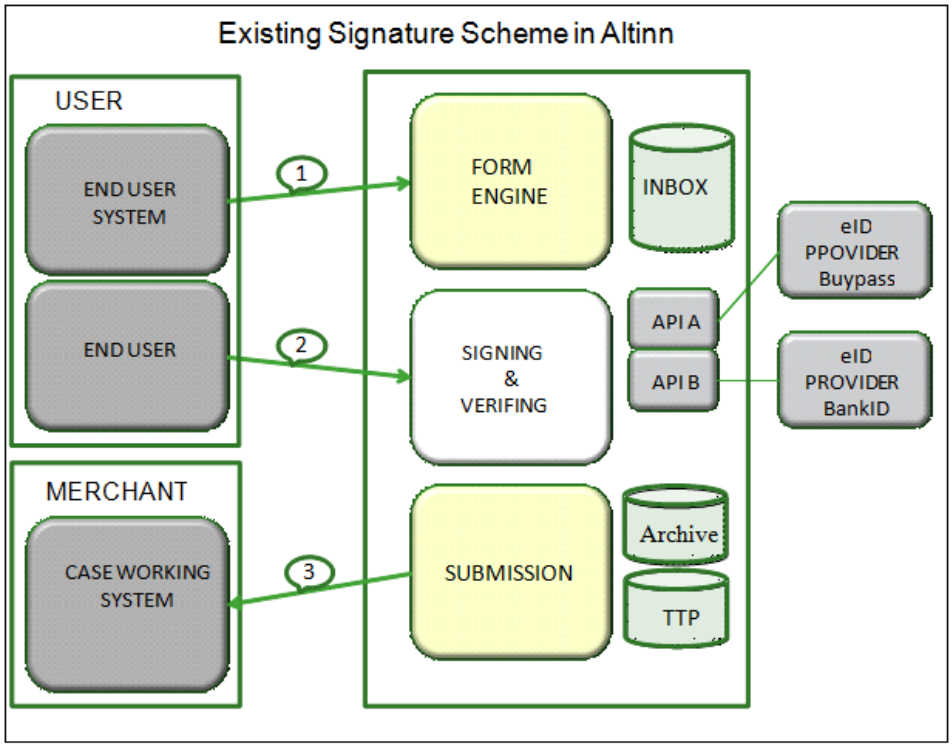

Figure 3.1: Existing Signature Solution in Altinn [20]

3.1.1 Technologies and Mechanism in Existing Soltution

This section presents the technologies utilized in the so called "login signature" maintained by Altinn and utilized by both Lånekassen and other public sectors in Norway.

Form Engine

One of the important technologies, used in the existing solution of the Altinn is form engine. Form engine make it mandatory for the signer to read what is to be signed. For example when a student wants to sign debenture paper he must read the policies of Lånekassen in order to understand what he is going to sign. The form engine persuades the signer to read before accepting the conditions because it is step by step form. The disadvantage of this engine is that it makes the signing process inconvenient [4]. This could be a reason for removing this engine in the new solution.

Singing and Verifying Mechanism

Altinn uses Buypass smartcard and MinID for both authentication and signature which is based on public key infrastructure (PKI). This is a currently available and the most reliable mechanism for signing with Altinn [76].

Public key infrastructure is framework consisting of policies defining the rules under which the cryptographic systems operate and procedures for generating and publishing public key certificate [50]. This technology is described in section 2.1.

As mentioned above Altinn uses PKI for signing and verifying of the signature. This makes the solution more trustable but there are still many flaws which are not considered in this section.

API of eID Providers and Applet

In order to connect Altinn with eID providers application programming interface (API) is needed. Altinn has integrated two eID providers in their existing solution, BuyPass and BankI ID.

API is a set of standardized predefined rules that a program can request upon. Almost every application depends on API of the underlying operating system to perform such basic functions as accessing the file system. With APIs, applications talk to each other without any user's knowledge or intervention, in addition make the application independent of the programming language [85].

Altinn in the existing solution has two eID APIs, BankID and BuyPass. For both signing mechanism Altinn uses applet, not an application. The difference between an applet and an application is that an application has main method while an applet does not. An applet is invoked from a web browser while an application must be run independently [85].

Archive and TTP archive

An archive is made for saving forms after signing. The signed forms by using Altinn will be archived in 10 years unless it is requested for not saving it.

In addition there is a Third trusted Party (TTP) archive where documentation of non-denial of submitted data is archived. Service providers can access this archive and it is possible to recreate how the service appeared to the user. There is a manual routine to extract data from third trusted party archive [4].

No form data is saved in this archive. Third trusted party archive, however, contain a collection of all logs made in connection with the submission of the form and the user, such as initiation, storage/modification of data, signing, forwarding, sending back to previous step, submission, opening of messaging and confirmation. These log information will be archived after the submission of the form/service [3].

SEID-SDO Siganture Format

Altinn uses SEID-SDO as described in section 2.5.2 for saving the signature as well as sending of it. It is a Norwegian standard developed in Norway and is extension of ETSI standards. This format was a preferred format chosen by Altinn in the existing solution.

For a common signing solution on a national level a SEID-SDO was recommended by Post and Telecom Authority in Norway [68].

The structure of SDO in figure 2.5 shows that many SDO seal can be put together. This shows that many signatures can be included in single object which solve saving of mutual signatures in long term [68]. As an example mutual signature, a contract consists of many dependent contracts signed by different firms can be saved in a single object where in one side a single firm or person is responsible and on the other side a number of firms thereby a number people are responsible for the contract. So the single firm or person must have signatures of all involved people or firms in the contract in order to be able to claim what was contracted [68]. This was the main reason that Post and Telecom recommended this format.

Web Services

Web service is an important part of web technology. It converts an application into a web-application, which further can publish the functionality of the application to end users [82]. Since Web Services platform is based on XML and HTTP where the latter is a transport protocol Altinn uses web services in communication layer. As mentioned above, the web services are used in communication between service provider and Altinn for transferring document in both ways. Description of this technology is given in section 2.2 and will not be repeated here.

3.1.2 Service Architecture

In this section a brief overview of services in Altinn signature solution is described.

Submission Services

Submission services consist of instantiation, filling out the form, validation, signing and submission.

Instantiation

Initiating is carried out by one of the parties where a control of what rights a user have will be done. A pre-filling data will be used if it exits when the form is instantiated [4].

Fill in the Form

A special component is developed in Altinn to show the forms according to ELMER-II standard.
ELMER-II-components have the following properties [23];

- Navigation menu
- Help menu
- Failure list menu
- Button for last/next page
- The form is in the middle
- A link for the general help as well as information about the form

In addition the service developer can choose to deviate from the mentioned list and add some other features for example.

- Controlled navigation
- Tracking option

In a form a calculation can be carried out. This calculation can be based on values in the field of the form which user is seeing or from the other side of the form. It is also possible to have attachments in the form. This form can be used dynamically i.e. one part of a form can be locked or opened based on the user's option. It is also possible to invoke web services for lapel outside of Altinn [4].

ELMER-II is standard concentrating on pedagogical challenges in the internet with the special focus on the forms. This standard do not take position on the content of the forms i.e. what information is requested but how to present the question in an online form context [23]. ELMER-II standard is formed to report the corporate messages to the authority in Norway [68]. Actually the specialist in large corporate do not fill in these forms but they are using professional systems. ELMER-II should serve the needs of the small company, where employees have a private person's relation to the forms. The form used in Altinn met all conditions given by the ELMER-II standard [4].

Validation

When the form is filled out, then the validation is possible for the service provider to denote that the form must be validated. The validation is consisting of field information whether the input type is correct, whether a check point that should be checked for example accepting or declining the terms and so on. The validation step refers the user to the part of the form that need correct input [4].

Signing

All forms are associated with workflow. Every form template regulates work process through the form's solution, for example a form can be denoted to have one or more signing steps. As mentioned above if the form requires signing it cannot be sent before signing is carried out. As well as a system developer can denote that parts of the form should be signed.

There are four options with using forms;

- No signing
- One signing step, with the same conditions as for authentication
- One signing step, with different condition for filling out and signing
- Two signing steps with different rights conditions [4].

The security level on each step can be managed by the service provider i.e. a service provider can put for example a security level 3 for offering a service to its customer. If a user does not have signing rights then user can send the form to the one who is authorized and has the signing rights.

Submission

The forms will be submitted when either all signing steps are performed or submission without signing is chosen. During the submission the following services are done;

- The worksheet is copied to the user's inbox (sent and archived submission)
- The worksheet is copied to the service provider's archive
- Submission information is logged in third trusted party archive
- The worksheet is deleted from the user's works list
- The user has the possibility of sending the receipt via e-post
- The worksheet is made clear for transmission to the agency if the agency has receipt batch
- The worksheet is sent directly to the agency if the agency has a direct receipt [4]

3.2 Difi's Proposed Siganture Solution

Difi is given the job to develop a common infrastructure for eID with a special focus on common solution for signing a document. Difi created a working group in order to work out the need for an advanced electronic signature. As mentioned above, a common solution for signing already exists in Altinn based on authentication and login. The working group concluded that there is a need for a common solution for advanced electronic signature in this context. There is no solution for an advanced electronic solution to be implemented. This common solution must make it possible for a user to sign a document appeared as a result of a dialog between service provider and a user. The important point is that the solution must satisfy the definition and specification of advanced electronic signature. Difi proposed two possible solutions that can be used to implement, synchronous and asynchronous signing scheme. In these solutions the working group tried to reuse most part of existing solution thereby the solutions are more Altinn focused. In this section a broad description and analysis of the proposed solutions are presented.

3.2.1 Synchronous Signing Scheme

According to [20] this scheme requires that a user must login to the merchant network through authentication portal, ID-porten. The document to be signed is created as a result of a dialogue between the service provider and the user and this signing must be performed in one of the steps in this dialogue which is called a synchronous solution.

This solution can be implemented by the merchant where signing service in Altinn will be used as a synchronous signing.

In this version as described in [20] pdf file can be used which probably PAdES standard will be used. Figure 3.2 shows high level system architecture of the proposed solution.
PDF document is sent to Altinn using web-services and it will be added to the Altinn user's inbox. The user is redirected to the Altinn signing page where the user can see the PDF document and choose an eID scheme for authentication, and then signs the document. user will get eID providers signing scheme, usually in an applet, to be used for authentication where the user must give security parameters for example pin code. Using web services the signed document will be sent back to the merchant and the document will be archived in Altinn. As in existing solution, non-denial information is logged in TTP. The user is redirected back to the merchant [20].

An important difference from the existing solution is removing of the form engine in Altinn. The use of form engine is no longer desirable as the working group concluded.

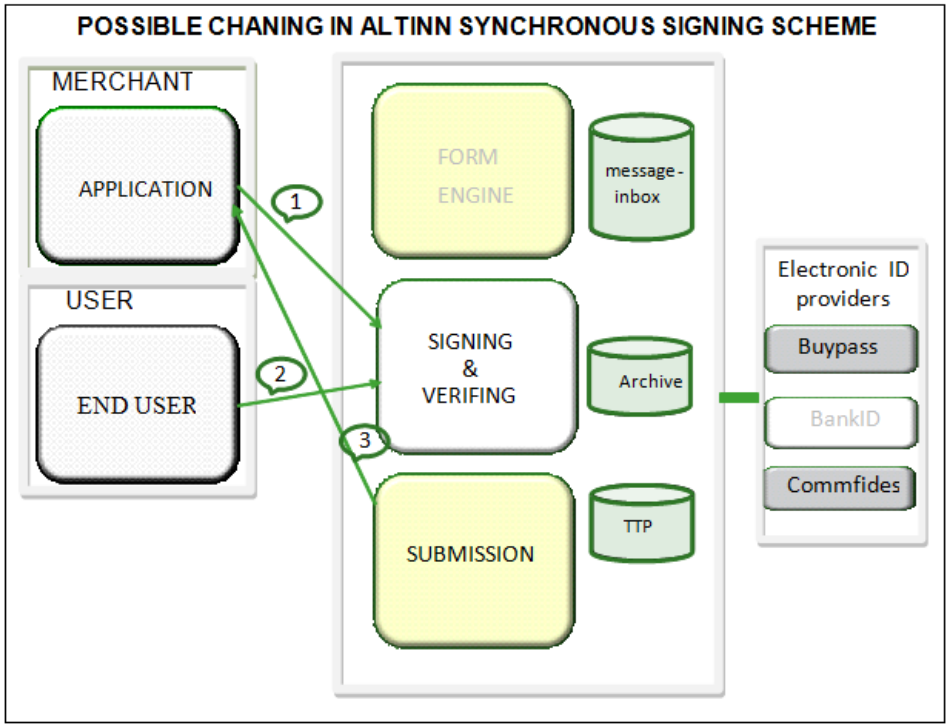

Figure 3.2: High level System Architecture of Synchronous Signing Scheme [20]

Service Architecture and Process Flow

Difi has proposed the so called synchronous signing scheme where some assumptions are made;

The solution is specified with use of asynchronous web-services i.e. the following steps;

- Merchant invokes web services in Altinn and sends the document and the corresponding parameters to Altinn.
- Altinn acknowledges receipt of document and the web service session is down
- When document is signed Altinn invokes the given web services of merchant and sends the signed document back to the merchant

Figure 3.3 shows the process flow of this mechanism.

Difi states some problems with this solution where Difi means that merchant must hear the incoming calls on defined web services. All merchants cannot offer such so-

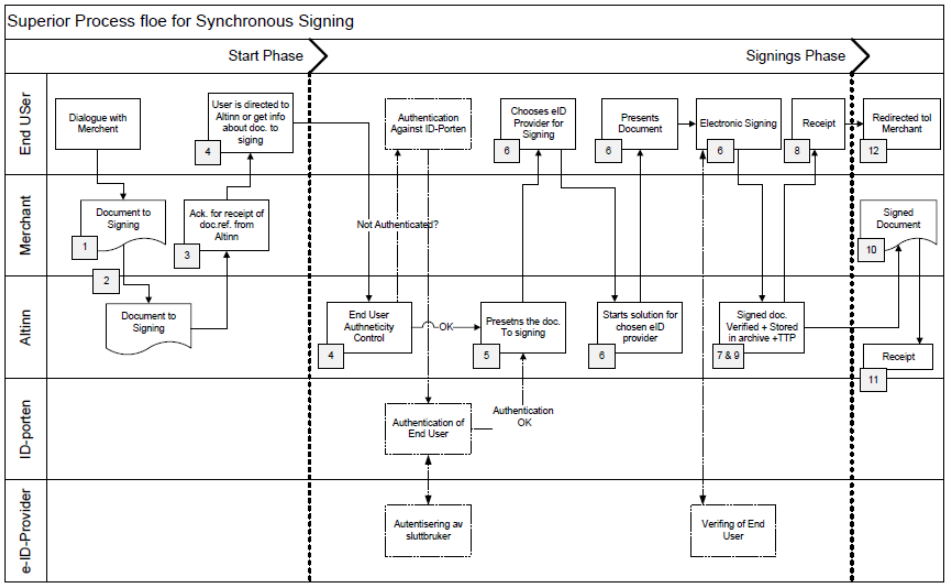

Figure 3.3: Process flow of synchronous signing scheme [20]

lutions because of some technical and security limitations. Other options according to [20] are as follows;

- Merchant invokes Altinn instead of reverse to obtain the signed document. This can be achieved by giving signal or merchant obtains completed documents regularly from the Altinn. A possible signal could be redirecting the user to merchant's website will mean that the document is signed
- Use of synchronous web-services where the originally session is waiting till the signing is completed is another option. Signed documents with the necessary parameters will be sent as response to the merchant originally invocation. The main problem as Difi states is timeout because reading a document by the signer can take long time

In synchronous signing mechanism, where asynchronous web services are assumed, a detail process flow is as follow;

1. As mentioned in the beginning of this section that a document arises as a result of dialogue between merchant and end user, when the user is ready to sign.
2. Merchant invokes web services of Altinn for signing service (SOAP over https) and sends the document with corresponding parameters to the Altinn. The parameters include document's identification, personal number of the user, URI for the redirecting of the user from Altinn, and another URI can also

be given where Altinn can invoke when signing is completed if asynchronous web service is used [20].

3. Altinn acknowledges receipt of document as response to web service call and returns the URI for redirecting the user. Then the SOAP session is decoupled.

4. Merchant redirects the user to the URI given as a parameter in the response from Altinn. The user is usually directed to the ID portal in order for the user to authenticate him before signing. Normally this redirecting mechanism is server side action where the user cannot see this step because of using of Single Sign on SSO technology. Redirection can also include the URI of the merchant in order to redirect the user back to the merchant. Meanwhile will Altinn check whether the end user is authentic?

5. When the user is authentic the document will be presented to the user.

6. The user will obtain the page for signing of given document. There are many alternatives that can happen. A user might be new user of Altinn then Altinn's consent page will be obtained. Since portal independent solution of Altinn will be used to approach the merchant's visual profile and merchant should specify prompts and eventually other content.

   The user chooses an eID for signing the document. Choosing eID depends on the user and an option could be MinID. The document is signed based on the chosen eID provider. An applet will be appeared for signing the document i.e. it would be an applet based solution. The document will be presented in order to see what the user is actually signing.

7. As decided earlier that Altinn must obtain some additional information to complement a Signed Data Object (SDO) for the document. This depends on the eID provider's solution. Altinn must ensure either by themselves or through eID provider's solution to;

   - Verify that signature (hash value) matches with the document to be signed
   - Check that eID used for authentication and signature is valid
   - Check that authorized person who signed is the one who claims to be i.e. whether the same personal number and eID is used both for signature and authentication
   - Add time stamp in the correct format [20]

   Altinn denied using SEID-SDO signature format due to that SEID-SDO is a Norwegian standard and in SEID-SDO a signed document cannot be read while this can be done using PAdES-LTV. Altinn cannot store the document and non-denial information longer than 10 years. As discussed with co-supervisor, Altinn's argument for not using SEID-SDO was satisfactory because this formatting is Norwegian specific which is not desirable, but PAdES-LTV as described in section 2.5.3 is chosen as a signature format

8. A receipt will appear to the user to ensure that signing was successful

9. Signed document is stored in an archive of merchant in Altinn. As in the existing solution the non-denial information is saved in TTP

10. Again the SOAP communication protocol i.e. SOAP over HTTPS, will be utilized for returning of the signed document to merchant. This occurs as Altinn calls merchant's web service for returning the signed document (the URI sent in the initial step will be used or it can also be configured in Altinn). Signed document with the corresponding parameters are sent back to the merchant.

11. Merchant acknowledges for the receipt document by disconnecting the SOAP session i.e. disconnecting SOAP session is a signal for document receipt

12. Finally the user is redirected to the merchant. Here again the URI sent as a parameter will be used to redirect the user to the merchant

If something goes wrong during the signing procedure for example the user cancel the signing procedure or internet disruption occurs and the document will be held in the Altinn user's inbox and meanwhile an error message will be sent to the merchant. Many errors can occur during these operations which are not stated here.

3.2.2 Asynchronous Signing Scheme

This scheme is the same as the synchronous scheme with a slightly difference as described here; In this scheme when the user is to be informed of the document to be signed there are two options;

- The merchant informs the user
- Altinn informs the user using the existing mechanisms in Altinn thus sending email or sms to the user

Another difference is when the user is informed about signing, the user can use two alternatives to log on and sign the document.

- The user can log on to the merchant where he will be directed to the Altinn signing page in order for the user to sign the document
- The user can log on directly to Altinn and select the document form task list and perform a signature based on the given policy

The same technologies are utilized in both solutions. A detail description of this scheme can be found in [20] and is not given here. The system architecture is illustrated in figure 3.4 below.

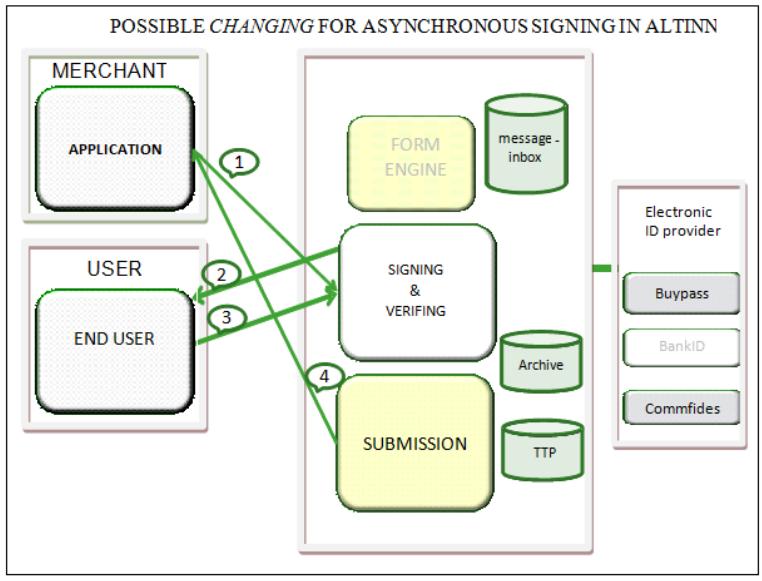

Figure 3.4: High level System Architecture of Asynchronous Signing Scheme [20]

3.3 Technologies and Standards to be Utilized in these Solutions

This section presents a review of the technologies to be used in Difi's pilot project of electronic signature. The main reason for reviewing the technologies is to discuss later why these technologies are preferred. Meanwhile the advantages and disadvantages of the technologies would be discussed. The technologies with more vulnerability are discussed in more detail.

3.3.1 Public Key Infrastructure

An important technology in digital signature is public key cryptography where asymmetric encryption is utilized. This technology assumed to be secure with specific usage. Difi intended to use this technology in the proposed solution. A brief review of this technology is presented in section 2.

3.3.2 Web Services

Difi will be using web services in their solution. Using this technology offers important advantages. Web Services are said to be modular, accessible, well-described, implementation-independent, and interoperable, reusable, deployable and simple [12].

Among others the disadvantages are as follows;

Simplicity is both an advantage and disadvantage because to be simple is good but this can in some circumstances be hindrance. Using plain text makes it simple to understand but the size of file increase i.e. the size will be bigger than encoded in binary protocols. System with low speed connection or extremely busy connections will suffer from this problem.

Using web service over HTTP and HTTPS will suffer from the stateless property of these protocols. When there is no data exchange the server may assume that the client is inactive and close the session and it will loose all information it was keeping. Handling of this problem is difficult for web services.

Since there are more advantages then disadvantages and the advantages are not fatal such that a system will suffer all the time choosing, this technology is a good idea.

3.3.3 Simple Object Access Protocol (SOAP)

As mentioned above that web services using SOAP for exchanging messages i.e. for communication as well as invocation of services. According to [65] SOAP is a XML-based protocol for messaging and remote procedure calls (RPC). SOAP works on existing transport protocol such as HTTP, SMTP, FTP, HTTPS, etc. There is no need for developing new transport protocol. Difi chose to use HTTPS to secure the content message.

Working of Soap protocol over any platform, any operating system, any programming language, in any computing environment and over any protocol makes SOAP heterogeneous and popular [29].

Besides the strength of SOAP there are weaknesses also such as big endian and little endian issues, packet size, implementation issues, security issues, versioning issues, message path, latency, no objects, reliability and trust, ontology and statelessness.

Among the weaknesses statelessness, packet size are described previously. A security issue is presented in detail here. Discussing all weaknesses are out of scope of this thesis and are omitted.

Security in SOAP is a big issue because SOAP cannot guarantee the security of the SOAP message. SSL is combined with SOAP to cover this shortcoming on network layer but securing content was not addressed. W3C addressed most of the shortcoming but could not succeed but to achieve security, trustworthiness the specification is brought to OASIS.

Even though OASIS dealt with those shortcoming but a weakness which can lead to breach of confidentiality is that the header of SOAP message envelop used to stick authentication data and unique identifier in order to find the right destination, but this data is sent in clear text which could be captured [29]. Here is the weakness SOAP cannot guarantee.

3.3.4 Electronic ID (eID)

Electronic ID (eID) is the most important part of this solution where authentication of the users done through eID. The security of the solutions relies on just eID whether it is secure or not. In order to build personal high security in a system approved eID must be used. Difi intended to use BuyPass, Commfides which have security level 4 while MinID deployed by Difi and is freely available for everyone in Norway has a security level 3 [19]. As a absolute requirement of the system eID provider must be registered with Post and Telecom Authority in Norway [68]. The named eID providers are registered and approved by Post and Telecom Authority. Using eID technology is an absolute requirement. This technology is described in detail in section 2.3.

3.3.5 SAML and SSO

Difi intended to use Security Assertion Markup Language SAML for exchanging authentication and authorization data between security domains in the proposed solutions.

A big question arising for why it is required, however there are many other technologies that can be used to implement. The main reason for this is the so called Single Sign-On mechanism such that;

- Limitation of browser cookies; using browser cookies, to implement SSO, do not support Cross Domain Single Sign On. To overcome this problem application of different technology is required
- SSO interoperability; using of the same SSO product is required for both sides in order to use SSO
- Web Services; as described in section 2.2
- Federation; instead of using a large variety of local identity management across organizational boundaries it is better to reduce to a single Federated Identity or at least a set of Federation Identity

SAML SSO standard overcomes these problems but there are still other problems which are discussed below.

In this solution Single Sign On (SSO) technology is intended to be used as used in the existing signing scheme. SSO is defined in section 2.5.1 but to be more precise

a concrete example is as follows;

If NAV is the first visited site and lånekassen is the new site. A user logs on to NAV where the user must authenticate using an eID to NAV, then the user wants brows to lånekassen, NAV will confirm the authenticity of the user to lånekassen.

SAML uses the so called SMAL SSO Browser Artifact profile described in appendix A for implementing Single Sign On.

Many flaws are detected in SMAL SSO Browser Artifact profile where it shows that this mechanism is vulnerable to three fatal attacks, man-in-the-middle attack, message replay attack and attack by information leakages [73].

As an example vulnerability of the SSL/TLS binding will be presented. As mentioned in Index A that SOAP over HTTP is one of the most important bindings of SAML SSO protocol. In this case Secure Socket Layer version 3.0 (SSL 3.0) or Transport Layer Security 1.0 (TLS 1.0) with a unilateral channel, where confidentiality and integrity of a message is desirable, is utilized.

Three attacks are launched by [73] where replay attack was overcome by challenge and response, for man-in-the-middle attack a big adversary is needed but the third attack by leakage of information is still possible.

As a conclusion this profile used for SSO in SAML standard is vulnerable to the latter attack. It means that SSL 3.0 and TLS 1.0 enhance the security but still cannot guarantee the security.

In addition to SSO Single Logout is also to be used. This mechanism makes it possible for the user to log out once and not needed to log out from every website.

3.3.6 PDF Advanced Electronic Signature - Long Term Validation (PAdES-LTV)

As mentioned earlier that SEID-SDO was intended to be used, but Altinn denied and they came with a new proposal of PAdES-LTV. This formatting support long term validity as the name shows. Since Altinn can only store documents up to 10 years¸ it was not desirable for lånekassen as debenture from lånekassen can last up to 30 years.

Difi's arguments for not using SEID-SDO were that SEID-SDO is a Norwegian specific standard while PAdES-LTV is international standard and the content of a signed document can be read in PAdES which is not possible in SEID-SDO.

A disadvantage of PAdES-LTV is that this cannot support different types of signature formats.

3.4 Analysis and Discussion of Difi's Proposed Solution

A broad study of both existing solution and the proposed solutions are carried out. The study shows the utility of new solutions. However there are a variety of entities and technologies in the existing solution that are reutilized. In this section architecture, standards and technologies of proposed solution are discussed, analyzed and compared in relation with the existing signing solution.

The existing solution uses the so called form engine. This form as mentioned is supported by ELMER-II standard where a signer is persuaded to read carefully before signing. This makes the existing solution inconvenient but at the same time more reliable. Passing through many stages will not be desirable for the end user which can breach the usability properties of the system.

New solutions omit this technology and prefer not to use it for the single reason that it was not desirable because of inconvenience.

When software is being developed and usability and security has to be considered, one will face that usability and security may come into conflict. When security is priority then often usability will be weakened because security makes the software more difficult to use [43]. The security in forms engine in existing solution was a priority which led to breaching the usability properties.

As mentioned earlier that the document will be sent to Altinn where the user should login and sign the document. The user who is one of the part in signing the document, will have the read and signing rights. This means that the user has no influence on document. By clicking on sign button the signature will be carried out. This breaches the definition of advanced electronic signature where one of the conditions is that the signer must be in possession of signing tools. In this case the signer has no control over the signing tools.

Signing and verifying is carried out by PKI. As mentioned the most reliable mechanism currently available is PKI. In both solution eID provider BuyPass is used where the BuyPass uses 1024 bit key which is quite difficult with the current technology to break. The most common algorithm used in PKI is RSA which is asymmetric algorithm, i.e. two different keys needed for encryption and decryption. This algorithm is based on modulus arithmetic. The security of this algorithm lies in the public/private keys pairs where breakings of these keys are not feasible because breaking these keys are factorization of prime number [78]. The longer the key length the stronger the security.

A recent research showed that a RSA modulus of 768 bits length has been broken where hundreds of machines from different research centers were used and the so called Number Field Sieve method was used. This process took several months.

To break this modulus using a simple computer will take more than 2000 years [74].

A 1024 bit key will make a large number consists of 300 digits where BuyPass is planning to increase the key length by 2015.

A research group from NIST has recommended that for protecting data within year 2015, a key of 1024 bits could be used. Protection over long terms will require a key of 1536 or 2048 must be considered [52].

Coming back to the proposed solution the key with 1024 bit length will not be adequate because an electronic signed document need to be saved within 30 years. We know how fast the technology evolves. A document with sensitive data content signed with key length of 1024 bits might be broken in 4-5 years [74], and the content of the document could be altered, signed again with the new content. Who will take the responsibility? If neither signature provider nor the signing parts take the responsibility then how can we trust the system?

Every public key is certified by a trustable certification authority (CA), where CA guarantees that the public key is valid and binds owner's identity to the public key [76]. CAs are physically entities located anywhere in the world who issues certificate. They may or may not be affiliated with governmental entity. How can we trust certificates?

Trusts make things difficult to believe in. Trusting certificate has the same issue. A certificate accoring to [51] can be trusted;

- *Directly* by exchanging the certificate by hand or calling the other part and asking him to read the fingerprint of the certificate
- *Hierarchically* where certificate is issued by a well know CA. When the CA is trusted, the certificate is automatically trustful
- *Indirectly* by sending the certificate by mail and checking the fingerprint on the phone, which might not always be possible
- By using *iSafeguard* model

On the other hand CAs use accepted standards for implementation of certificate [72].

In this context hierarchical is preferred since other ways of trusting is impossible. Altinn for authenticating itself uses SSL certificate. The figure 3.5 shows the certificate used by altinn [4].

The figure shows that a hierarchical trust model is utilized in order to trust the certificate. This model makes us sure to trust the certificate and thereby trust the public key and finally the signature.

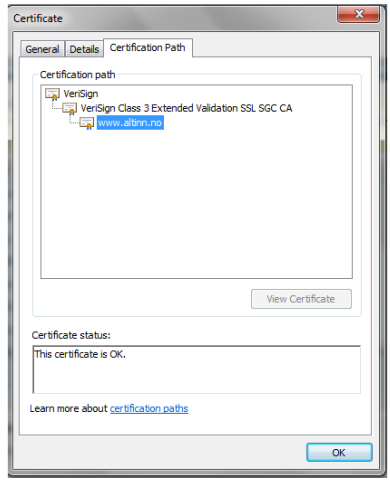

Figure 3.5: SSL Certificate Utilized by Altinn [4]

API and applet are common in web technologies where applet is used when extern functionality is desirable without implementing in the web. Using Applet establishes the "what you see is what you sign" problematic. There have been a lot of research done in this area which will not be discussed here.

Altinn use API for connecting to eID providers. Altinn has integrated API of BuyPass and and BankID. In the new solution BankID is drawn from the architecture. BankID could not fulfill the conditions given by Difi where BuyPass and a new eID provider commfides showed that they can add those functionalities [20].

Altinn stores a copy of document as evidence in its archive for 10 years where merchant can access this archive. If document is a debenture where down payment time is 30 years, what will happen if the debtor denies paying back the loan and altinn has deleted both copy of the signed document and non-deniable data after 10 years. Of course the creditor will have a copy of this signature but it would be difficult to trust whether the creditor is claiming the correct amount or not. Altinn proposed PAdES-LTV (described in section 2.5.3) formatting standard to fulfill this condition. This is a European standard which is better than SEID-SDO which is a Norwegian standard. This can be a weakness of altinn not to have the evidence after 10 years.

TTP archive saves the logging information of the users done during the signing processes and even going forth and back of the document, canceling the signature first time is logged. Logging information could be of importance if the debtor tried to deceive the creditor when he was signing the debenture paper. Through finding the probability of going forth and back while signing is indicating that debtor could have been trying to deceive the creditor.

As mentioned SOAP protocol is used for exchanging messages between entities, to be independent of particularly programming language and it is sent over HTTPS in order to protect it from any kinds of attacks.

Altinn uses SSL certificate for authentication of register unit in Brønnøysund, and for protecting the data exchanged between Altinn and other parties Transport Layer Security (TLS 1.0) is utilized. In this security RC4 for encryption with 128 bit with MD5 for message authentication are used where RSA with a 1024 bit key length for the key exchange. Figure 3.6 illustrates certificate utilized by Altinn.

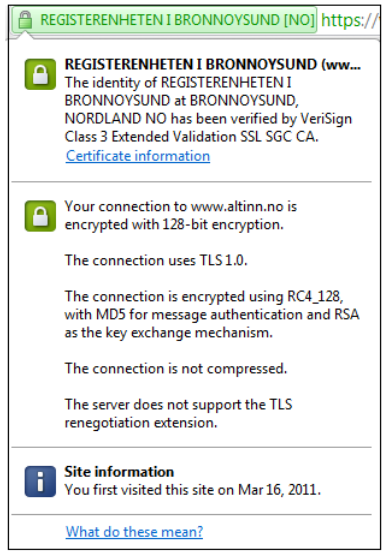

Figure 3.6: Information of SSL Certificate utilized by Altinn

RC4 with 128 bit key is not quite strong neither MD5 is supposed to be a strong hash algorithm but RSA could be strong enough with currently key length of 1024 bits. According to [84] MD5 is already broken. As described in [69] that RC4 is vulnerable to some attacks especially when the beginning of the output key stream is not discarded, or related or non-random keys are used. Using of RC4 can in some circumstances with a special usage, lead to very insecure cryptosystem, for example WEP uses RC4 for confidentiality which has advantages of simplicity and speed in software but it could be broken within few seconds.

While Altinn uses $RC4_{128}$ with MD5 and RSA as security parameters in transport layer, Difi in ID porten uses a stronger algorithm Advanced Encryption Standard in Cipher Block Chining AES-CBC with 256 bits key length and SHA1 as a hash algorithm where RSA is used for key exchange.

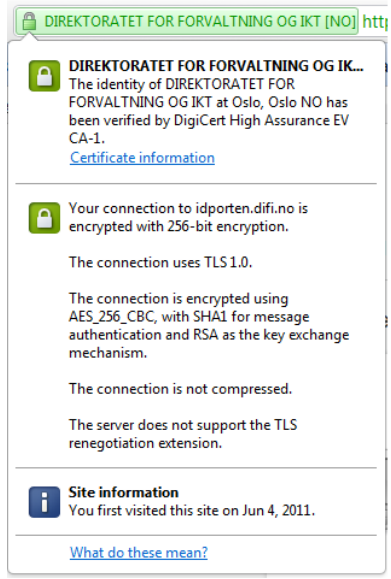

Figure 3.7: Information of SSL Certificate utilized by Difi i ID-porten

ID portal uses stronger security in transport layer than Altinn. Difi has thought that the user will be directed to the ID portal for authentication from merchant or Altinn, in the new solution. They further meant that Difi with ID-portal will be responsible for authentication and Altinn will be responsible for signing. Difi uses longer keys for exchaning encryption key. Figure 3.8 illustrates the key information of certificate utilized by Difi.

Altinn in both existing and new solution uses SAML for asserting security in the messages in XML format. An advantage of this standard as mentioned earlier is that Single Sign On mechanism can be implemented easily using this standard. Another reason for using SAML is that it supports protocol binding that embeds SAML construct for transport. Using binding mechanism will give an advantage of not to develop new protocols for communication.

Despite all these advantages of utilizing SAML it has also weaknesses especially when SAML SSO Browser Artifact Profile is used for implementation of SSO. As mentioned that the man in the middle and replay attacks overcame but it is still information leakages attack. This can lead to some vulnerability of the system.

Despite asserting security to the transport layer SAML is vulnerable to information leakage attack. Adding SSO to the signing service will make the solution vulnerable to information leakage attack.

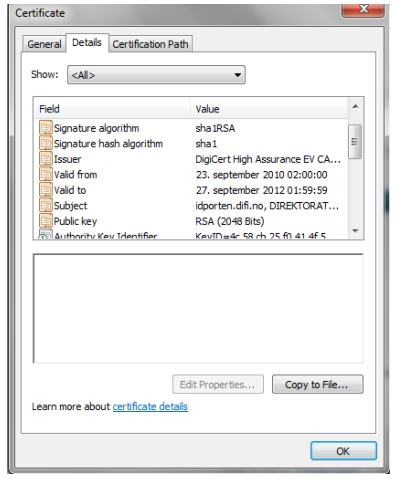

Figure 3.8: Certificate utilized by Difi in ID-portal

Chapter 4

Proposed Solutions Based on Proxy Signer Concept

"*The mantra of any good security engineer is: 'Security is not a product, but a process.' It's more than designing strong cryptography into a system; it's designing the entire system such that all security measures, including cryptography, work together.*"[9], Bruce Schneier

Proxy signature is a signature concept that a person (original signer) delegates its signing capability to another person (proxy signer) in order to sign on behalf of him/her. This concept is first studied by Mambo et. al .in late 20th century [42]. This idea is described in detail in section 2.4. This section presents proposed solutions of this thesis in detail.

Three solutions are proposed where the proxy signer concept is used in all of them. In the first and second solutions service provider/user model is used where in the first model, user centric, the users are involved more than the proxy signers i.e. proxy signers are not talking to each other, while in the second solution, proxy signer centric, proxy signers are involved more than the users i.e. the proxy signers talking to each other. Finally the third solution where two private people are signing a document online is discussed.

This section includes requirements of the proposed solutions,

4.1 Entities of the Proposed Solutions

This section presents the entities involved in the system. All presented entities in this section are common in proposed solutions except Mail Archive which is utilized in the last proposed solution.

4.1.1 Electronic ID Provider (eID)

Electronic ID provider in this model is the main trusted party where the system relies on. In Norway there are two eID providers, buypass and Commfides, who issue smart cards and MinID provides authentication of users and is maintained by Difi. According to eSignature law authentication with smart cards has security level 4 while MinID has the security level 3 [19]. Electronic ID provider issues electronic IDs which is described in more detail in section 2.3 which will not be repeated again here.

4.1.2 users

users in this model are people who usually sign documents online using the signing application on their computer. These applications are talking to each other and with their proxy signers also. It is assumed that these users' computers have the capabilities of issuing keys to their proxy signer as described in section 2.4.1 and creating authentication data as described later in this section and creating signing order. users in one of the three proposed solutions are active in signing and verifying the documents while in the second and third proposed solutions users are passive in the signing and verifying the documents. The services a the user entity (user computer) can perform are described below.

Creating Signing Order

First a warrant message (containing personal information of the original signer and part of the message) is created. The same warrant message is used for both creating authentication data and signing order. This message is concatenated with proxy signer's public key and hashed. The hashed value is signed by the original signer using its own private key. Along with the signing order warrant message will also be sent to proxy signer. The confidentiality of personal information is protected by using secure channel created between user and proxy signer using SSL/TLS certificate. The value of signing order changes every time an order sent. The signing order creation is illustrated in figure 4.1 below.

Creating Authentication data

As mentioned above that the same warrant message will be used as used for signing. A digest SHA1 of proxy signer's private key (X_p derived by the user) concatenated with warrant message is computed. The hash is signed by the original for signer (user) using original signer's private key (X_u). This piece of signature is used for authenticating him to a proxy signer. The procedure for creating authentication data is the same as in creating signing order, but different keys are embedded in the signature before signing. Figure 4.2 illustrates the creating authentication data.

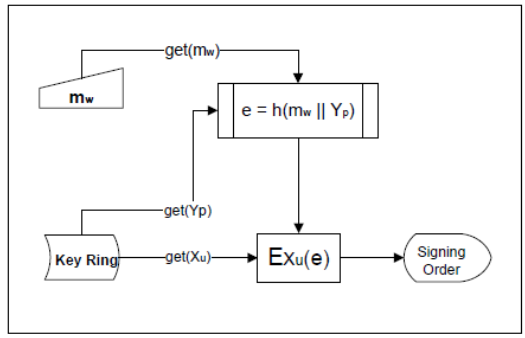

Figure 4.1: Creation of Signing Order by user(Original signer)

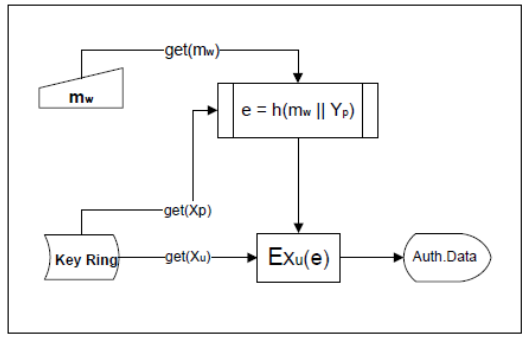

Figure 4.2: Creation of Authentication Data by user (Original Signer)

4.1.3 Proxy Signers

A Proxy signer is meant in this thesis to be a server signing a document on behalf of a user who has given his signing capability to it. The important fact behind this server is how to maintain this server in order to achieve both trust among the users and how to develop it. A proxy signer server will have the capability of both signing, formatting and validating a signature. This server is connected to a database maintaining copies of document signed and validated by it. Services a proxy signer can provide are discussed below.

Authentication of user to Proxy Signer

When a document is received by proxy signer it will first check the authenticity of user whether the user is authentic and whether user has delegate its signing capability to the proxy signer.

As discussed in section 2.4.1 when private key is issued to proxy signer the key will be transferred in secure manner to proxy signer. Along with the key, proxy

signer's private key is hashed and signed by original signer which will be used as pre-authentication data in subsequent authentication of original signer to his proxy signer. Every time original signer wants to sign a document, this pre-authentication data will be sent along with the document for authenticating of original signer. Proxy signer will compare the received pre-authentication data with the pre-received pre-authentication data. If these are identical the user is authentic.

This has a weakness if this data is captured by a third non-desirable party it could be used to authenticate a fake person as a valid person to proxy signer. To overcome this problem a warrant message (as described in over) will be used where proxy signer's private key concatenated with a warrant message is hashed and signed by original signer. In every authentication a new warrant will be used, of course the subsequent documents for signing will not be identical. Every time the warrant message will be sent along with the authentication data. Proxy signer will concatenate his private key with the warrant message and hash it using the same hash algorithm used by original signer. Proxy signer decrypts the received authentication data using original signer's public key and compares it with the hash computed by proxy signer. If the received hash and computed hash are identical the user is authentic otherwise the user is a fake person.

The advantage of this mechanism is that the shortcoming discussed above is addressed and the disadvantage is extra computation of hash of warrant message with the proxy signer's private key by proxy signer. The figure 4.3 below illustrates the authentication of original signer (user) to proxy signer.

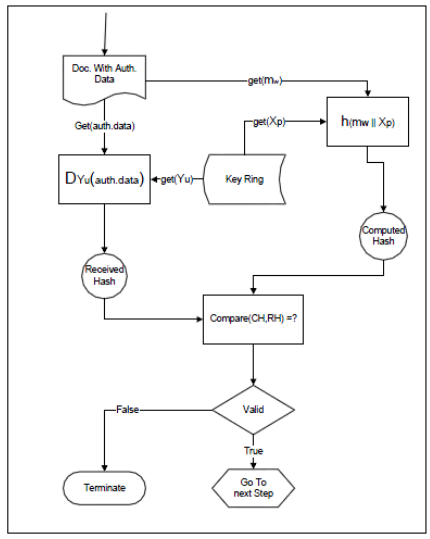

Figure 4.3: Authentication of user to Proxy Signer

Validating Signing Order

The signing order is checked by proxy signer by computing the hash of warrant message concatenated with proxy signer's public key. The received signing order is decrypted using original signer's (user) public key. The decrypted hash value is compared with computed hash value. If these are identical the signing order is accepted otherwise the document is marked as invalid document. This will protect coping signing order and sending to the proxy signer by a fake person. The figure below illustrates validation of signing order. It is assumed that the certificates are valid, and are not considered in the figure.

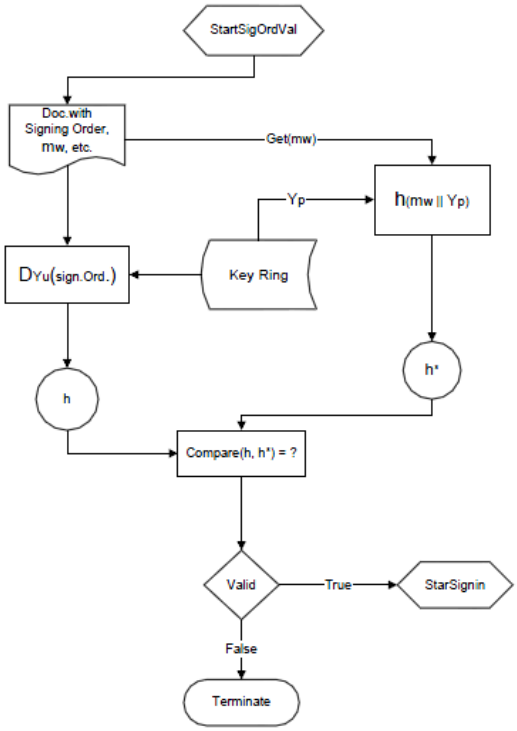

Figure 4.4: Validation of Signing order by Proxy Signer

Singing a Document

When user is authenticated the proxy signer will sign document. Figure 4.5 below illustrates how signing is carried out.

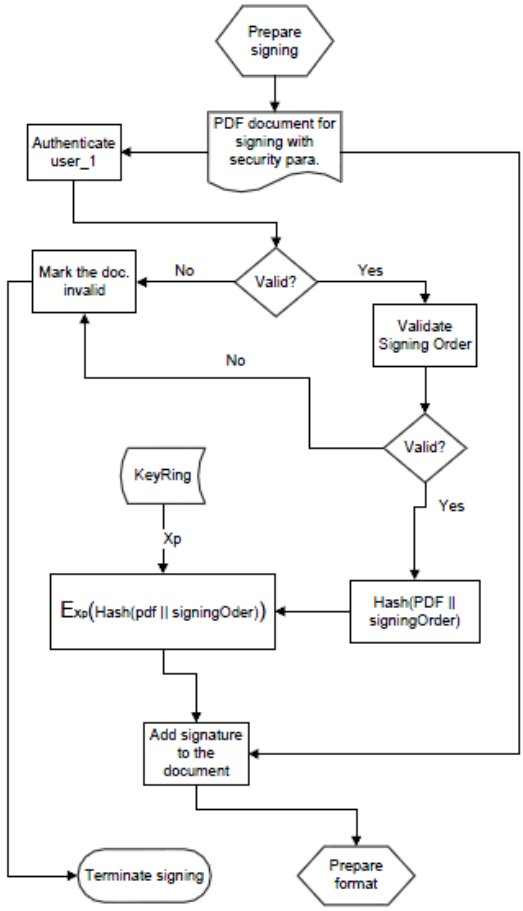

Figure 4.5: Signing a document by Proxy Signer

Since the proxy signer receives signing order along with the document it will concatenate singing order with the document and compute hash of them using SHA1.

Adding signings order with document to compute a hash has an advantage that the original signer cannot deny that he sent the order. Putting signing order will secure the non-repudiation as well as the modification properties of the signature. If document is modified the hash of document with the other two parameters will not be identical.

When the hash is ready for the signing, proxy signer will retrieve its private key, sent by original signer (user) previously, from key ring and sign the computed hash. The key with a unique ID related to the original signer is stored in proxy signer's

key ring. This key is issued by the user by driving it from his private key (stored in the eID smart card). The signature is then added to PDF document and sends it to formatting.

Formatting

When the signature is carried out it must be formatted according to standards in order to save it for long time and send it over a communication link.Figure 4.6 illustrates the process flow of formatting the PDF document after it is signed.

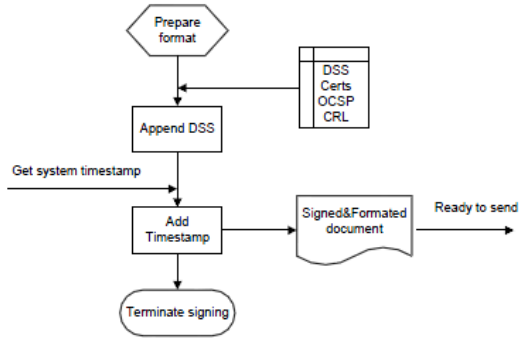

Figure 4.6: Formatting the document after the signing phase

As mentioned in section 2.5.3 that PAdES-LTV standard will be utilized saving and sending the document back and forth between users and proxy signers. In PAdES-LTV profile Document Security Store (DSS) is first appended to the document and then add all related validation data which will be required for eventually validation of signature. An optionally Validation Related Information (VRI) can also be included where indirect references to a specific signature are stored. DSS is a dictionary where it has four entries with keys VRI (of dictionary type), Certs (of array type), CRL (of array type) and OCSP (of array type).

All certificate, , including proxy certificate which is connected to the proxy signer's private key used for signing the document, related to the signature will be stored as stream in array with Certs entry. This can make the implementation easier as well. When checking for certificates it will be a look up in the array to retrieve the desired certificate. In next entry which also is of array type containing a stream of OCSP response. In the last entry, which also is of array type, a CRL as a stream will be added.

An optional dictionary entry inside the DSS dictionary exists which contains signature VRI dictionaries in the document. Entry key of this dictionary is encoded in Hexadecimal (uppercase) and is SHA1 digest of the signature to which it applies and the value is signature VRI dictionary which contains validation related

information for that signature [25]. In addition to CRL, OCSP and Cert entries VRI dictionary has two more entries, TU time and date of the creation of VRI and Time Stamp (TS) which do the same job as TU. One of them should be used in implementation. An entry in this dictionary can be used for adding miscellaneous security parameters. Miscellaneous security parameters desired are signing order, as described in section 4.1.2, and authentication data, as described in section 4.1.2.

The structure of DSS is described in section 2.5.3, for more detail see [25].

When all security related information is added to DSS, time stamp to the document will be appended. The structure of Document Time-Stamp is described in section 2.5.3. This Document Time-Stamp is an ISO 32000 extension which has a standard dictionary structure with some changes. The dictionary has the following entries;

- *Subfilter* which identifies the format of the data contained in the stream. It might be necessary for the reader to read when conforming signature handler can read a specific format. Value is left for the developer to decide what to save in this field
- A content entry called *Contents* of byte string type exists for representing the value of the byte rang digest
- An entry called *V (version)* of integer type representing the signatures dictionary format

All timestamp information relating to the signature is added in this dictionary.

Verification

A proxy signer must also be able to verify a signature if desirable. User in this model must be able to verify a signature. The model will be described here when proxy signer is verifying the signature but it is also valid for user which could be able to validate a signature. Figure 4.7 below illustrates the validating mechanism.

When Proxy signer receives the document it will start security approval from the outer part of the document namely from the document time stamp. It will first check timestamp of the document in relation to current system timestamp. The proxy signer will check the timestamp against current time and if the timestamp is valid the document will be forwarded for further consideration if not the document is marked as invalid.

If the timestamp is valid then verifier (proxy signer/user) will extract DSS where certificates will be retrieved. Before validating the signature verifier, proxy signer or user, will check the validity of the certificates used in the signature. Certificate validating methods are described in section 2.1.1. If at least one of the certificates is either revoked or invalid the document is marked as invalid document. If not, the verifier will start signature validation as follows;

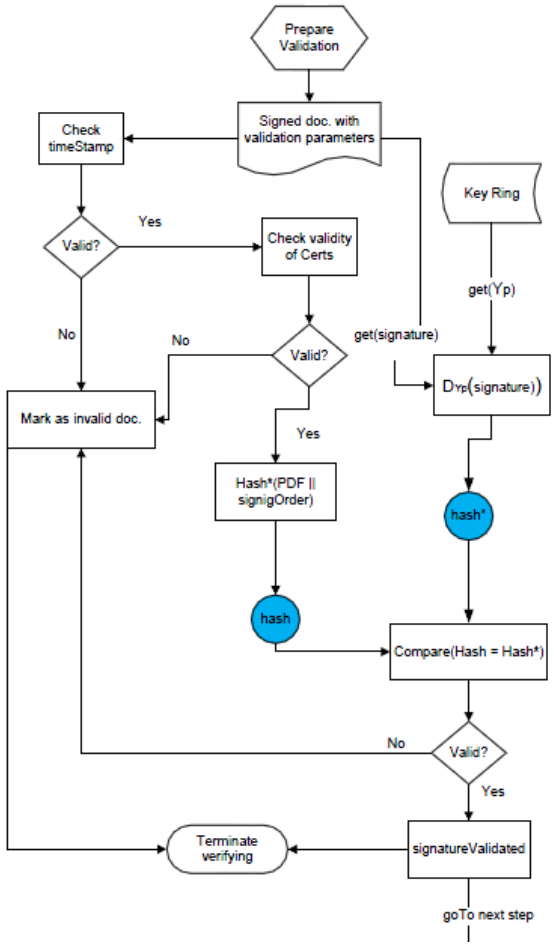

Figure 4.7: Verifing the Signature

1. Concatenating the document with the signing order sent with the document and compute a hash using SHA1

2. Extracting the signature of the document and decrypting it with proxy signer's public key (derived from original signer's public key as described in section 2.1.3)

3. Comparing the decrypted hash (signature) with the computed hash, if they are identical the signature is accepted otherwise mark the document as an invalid document.

When the signature is validated either by user or by the proxy signers it will be sent for further actions.

4.1.4 Mail Archive

In the third proposed solution where an extra entity is used called Mail Archive. This entity is utilized as mailing system where one can send and receive mail. Altinn, digipost, normal mailing system and so on can be utilized as a mailing archive. Altinn as described earlier is a national internet portal which could be used for different purposes such as getting and signing tax paper [4]. Digipost is newly digital post system where one can get post digitally. Digipost has lifelong archive to save the document while Altinn can archive the document in 10 years. This system is connected to Norwegian population register where every user is checked against this register [21]. This could be an advantage of this system but Altinn has TTP as described in section 3.1.1 where non-denial information is saved there, which digipost does not have.

Sending post to digipost is carried out as sending post home. It is not like mail systems where one can both send and receive mail freely, but in this system one can just receive post freely not send. The sender of post must bear the postal charges.

In this context in third proposed solution it would be used as a mail archive where PS_1 can send signed document to U_2 and PS_2 can send to U_1.

4.2 Communication channel between entities

Channel between U_1 and U_2 in both user centric and proxy centric is not encrypted in the beginning, normal HTTP request/response method is utilized, but when they have agreed about a service, U_1 is redirected to authentication page of U_2 the channel is encrypted. In this channel HTTPS with Redirect/Post methods are used where the confidentiality is achieved through SSL/TLS security mechanism.

The channel between U_1 and eID provider, in all solutions, is a front channel and there is also a back channel between U_2 (Service Provider) and eID provider 1st and 2nd solutions, but there is no back channel in the 3rd solution. In the 3rd solution there is a back channel between eID provider and Mail Archive. In the front channel SAML over HTTP/S could be used where in back channel SAML over SOAP over HTTP/S could be utilized. As described in section 2.5.1 that SAML standard utilizes envelope method to achieve assertion of security in message. SAML message is enveloped i.e. embedded in the body of SOAP message where latter envelope is again enveloped i.e. embedded in the body of the HTTP message.

Sending document to U_1 after authentication could be carried out through secure channel for preventing altering in confidentiality, authenticity and integrity properties of the document. Since both sensitive parameters are included in the document it must be protected, because changing for instance the address the signed docu-

ment to be sent to will lead to a large conflict between U_1 and U_2.

The channels between users and their proxy signers in 1st and 2nd solution, and the channels between users and Mail Archive, SOAP over HTTP/S could be used to secure the channels.

The challenging parts of communication channels to be implemented are between PS_2 and U_1 PS_1 and U_2 in the 2nd solution. The best ways to achieve security in these channels are to use SSL/TLS. When for instance PS_1 sends a link to U_2, giving limited access to $PS_1's$ file archive; before U_2 can download the document a secure channel must be created.

The secure channel is created as follows; when U_2 clicks on the link sent by PS_1, $U_2's$ browser requests $PS_1's$ server to identify the server. PS_1 sends a copy of its SSL certificate. $U_2's$ browser checks whether it trusts the certificates, if so the browser sends a message to $PS_1's$ server. The PS_1 sends a digitally signed acknowledgement and starts an SSL encrypted session [75], [76]. The key exchange and encryption and decryption are not described here.

4.3 User Centric Proposed Solution

In this architecture the users are actively interacting with the each other as well as with proxy signers of each other, but the proxy signers do not talk to each other. An eID provider is the central trust party that both users trust on.

4.3.1 Sytem Architecture

In this solution asynchronous web services are assumed to be used. A mutual signature is carried out. User$_1$ (called U_1 in the rest of report) is not logged in to User$_1$'s web (called U_2 in the rest of report). For the sake of simplicity of figure error handling are not considered here. The dialogue between U_1 and U_2 are carried out in advance. The system architecture is illustrated in figure 4.8

A document as a result of an agreement between two $U_1$and U_2 is created where in this model U_1 is a service user and U_2 is a service provider. This is assumed to be an agreement about a service. U_1 requests for service from U_2 (1) and U_1 is directed to eID-provider for authentication (2). U_1 authenticate himself he will get the document for signing (3). Authentication data sent direct to U_2 in a secure back channel. U_1 reads the document and wants to sign where he sends the document to his Proxy Signer$_1$ (called PS_1 in the rest of report) (4). PS_1 checks authenticity of U_1, and if U_1 is authentic PS_1 signs on behalf of U_1. PS_1 sends the document to U_2 (5) where U_2 validates the signature and sends it to Proxy Signer$_2$ (called PS_2 in the rest of report) (6) with a signing order. PS_2 checks the authenticity of U_2, as PS_1 did, and signs the document on behalf of U_2. PS_2 saves a copy in its

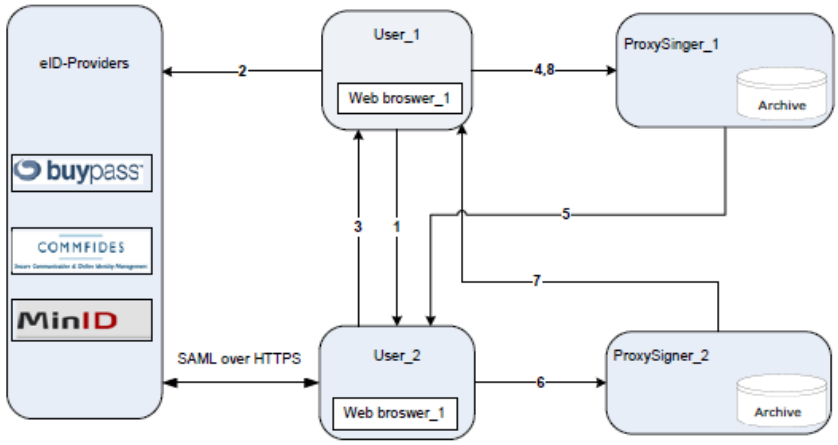

Figure 4.8: System Architecture of the Solution Proposal

archive for the later use and sends (7) the signed document to U_1. U_1 validates the signature and approves the content of the document by comparing it to document sent in the first step. If document is approved a copy will be sent (8) to PS_1's archive for saving for the later use.

4.3.2 Process Flow

In this architecture asymmetric web services are assumed where U_1 invokes its Proxy signer's web services to send the document to. On receipt of document by proxy signer the SOAP session is down. A dialogue between U_1 and U_2 has been carried out where they have agreed to sign an agreement. This dialog and error handling are not shown in figure for the sake of simplicity. Figure 4.9 illustrates process flow of this model.

1. Electronic ID provider will authenticate U_1. This could be achieved using an applet connected to eID provider, but if MinID is used for authentication U_1 will again be redirected to ID-porten (MinID server) for authentication. The security level for authentication is 4 (personal high) for using an approved eID provider. Meanwhile all authentication information, included time stamp, will be recorded with eID provider according to the eID policy. Succession of authentication will also be acknowledged to U_1. Directing U_1 back to U_2's web page will indicate that the U_1 is authenticated. Using eID provider with an applet solution, will also indicate the succession of authentication in the applet that the "authentication was successful".

2. When U_1 is authenticated U_2 will send document to U_1 for signing. This is achieved through directing U_1 to file archive where U_1 is given limited access

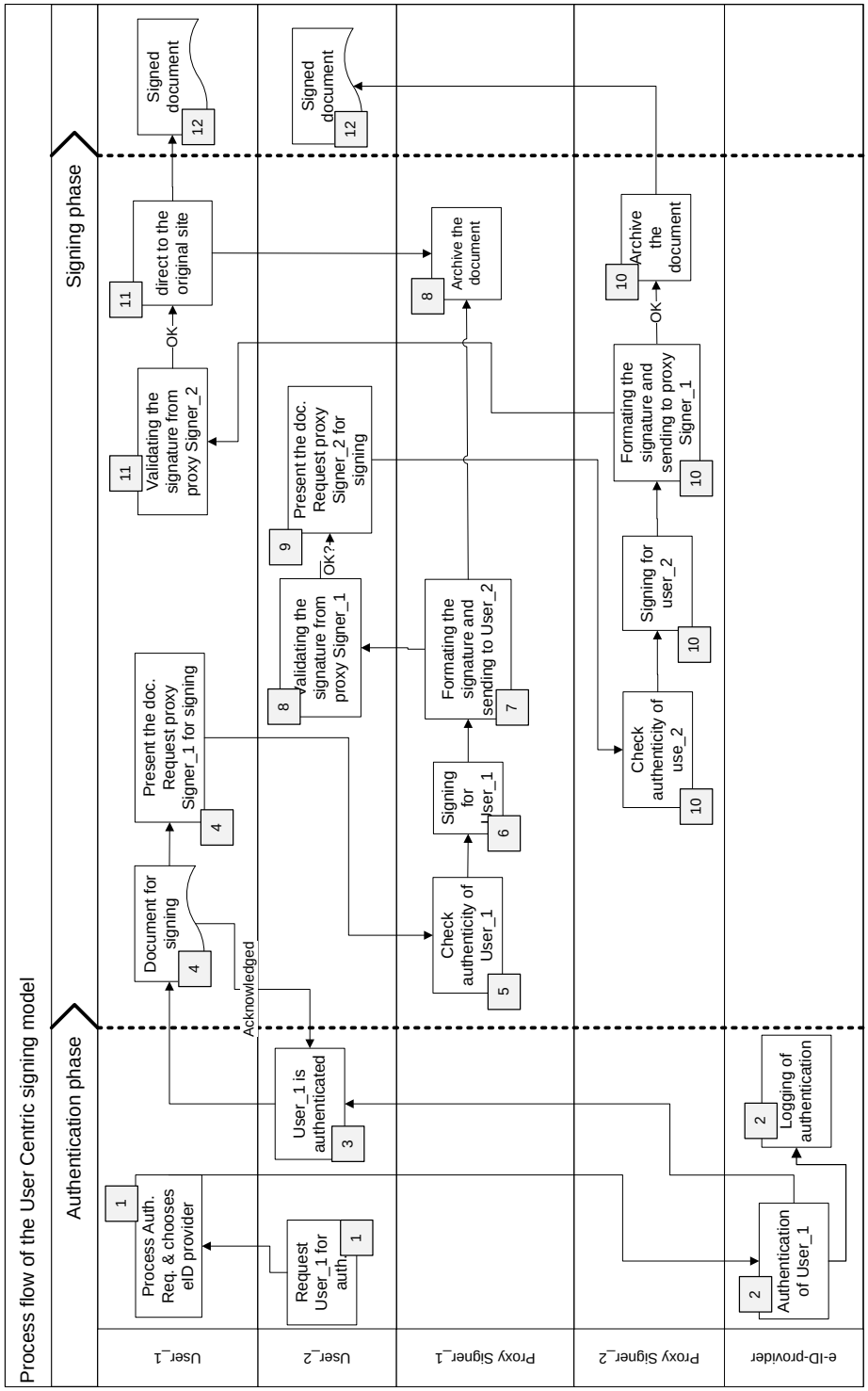

Figure 4.9: Process Flow of the User Centric Signing Solution

rights to this archive, and U_1 will download the document to his own machine for further consideration. In other words, the document is in U_2's server where U_1 is granted limited access through a secure channel to download it. Secure channel is provided using SSL/TLS certificate.

Alternative is to include Mail Archive in the solution where to just collect the document via Mail Archive as proposed in third proposed solution described in section 4.1.4. This could be a better solution because this facility already exists, but a disadvantage is that system is getting dependent on Mail Archive which is not desirable. This is the main reason that Mail Archive is not included in this solution.

Along with the document the corresponding parameters will be sent to the U_1as well. These parameters includes personal information of U_2 i.e. service provider (organization number prefixed with land's code, name and id of the case worker who deliver the service and so on), URL where the signed document to be sent to, a unique document ID which can be used between U_1 and U_2, name of the document and what the document is about, deadline for downloading (this could be few minutes otherwise timeout will kill session which can lead to re-authenticating the user), options for choosing authentication level and what levels can be accepted and so on. A complete parameters list is dependent on implementations.

3. U_1 has a document for signing where U_1 retrieves the document and reads the content of the document. Upon acceptance of the contents of the document U_1 invokes his proxy signer's web services to send the document with the corresponding parameters. The parameters included are U_1's personal information and his authentication data (as describe in section 4.1.2 how to create authentication data), URL(i.e. U_1's mail address where the link for downloading the document to be sent) received from U_2 for where to send the signed document, what security level is needed for signing, whether the signature to be saved in proxy signer's archive if yes there will need some extra information for example minimum required validity, whether the log of action performed to be send back to user of the proxy signer and many more according to different session type. Along with these parameters, a signing order in cryptographic form, as discussed in section 4.1.2, is sent to PS_1 which will enable the proxy signer to sign the document. For transferring the data between U_1 and PS_1, SOAP over HTTPS will be utilized. This will secure the content of the document in the link between U_1 and PS_1 which achieves the confidentiality properties of the document. When PS_1 receives the document it will acknowledge the receipt of document. SOAP session will be decoupled.

4. When the PS_1 receives the document it will first check the authenticity of U_1. This is achieved through checking the authentication data as discussed in

section 4.1.3. The confidentiality of the document in the link between $U_1$and PS_1 is again secured by SSL/ TLS certificate.

5. PS_1 signs the document which is described in detail in section 4.1.3. In this step the signature is created here where some security threat and attacks are also considered .

6. PS_1 formats the document with the signature and prepares sending it to the U_2. In formatting PAdES-LTV standard will be used where this standard has many advantages over for SEID-SDO which Altinn is using in the existing solution. A copy of the signed document will be archived in $PS_1's$ archive till the mutual signed document is received by the PS_1. This is to prevent the modification of the document by U_2. The formatting of the signature is discussed in more detail in section 4.1.3. PS_1 uses the URL (sent in step 4) to send the signed to U_2.

7. When U_2 receives the signed document it will first check the validity of document timestamp by extracting the document and checking the timestamp as well as the validity of certificates of Certification Authority (CA) certified the PS_1 certificate and the certificate of the CA who certified the public key of U_1. This is achieved by checking the certificates using Online Certificate Status Protocol (OCSP) or Certificate Revocation List (CRL) invocations. If these certificates are valid then U_2 will validate the signature which is discussed in detail in section 4.1.3.

8. After the signature is validated U_2 requests PS_2 to sign the document. U_2 invokes the PS_2's web services (again SOAP over HTTPS) to send the document to its proxy signer where the same technologies described in step 5 are utilized. PS_2 acknowledges the receipt of the document which leads to decouple the SOAP session. Along with the document some parameters are sent to PS_2. The parameters included are U_1 and U_2's personal information, URL received from U_1 for where to send the signed document, what security level is needed for signing, whether the signature to be saved in proxy signer's archive if yes ,it will require some extra information for example minimum required validity, whether the log of action performed to be send back to user of the proxy signer and many more according to different session type. Along with these parameters, a signing order in cryptographic form, as discussed in section 4.1.2, is sent to PS_1 which will enable the proxy signer to sign the document.

9. When PS_2 receives the semi-signed document the same procedures carry out as in step 5, 6 and 7. PS_2 sends the document to U_1 after formatting the document. A copy is saved in $PS_2's$ archive for the later use.

10. When finally mutual signed document is received by U_1 he will validate the signature as done by U_2 in step 9. Sending the document to U_1 is achieved using the same procedure as in step 4 where U_1 is informed by sending a link

to download the document from U_2's file archive.

After the document is validated and approved by U_1 it will send a copy of the document to PS_1 for archiving for later use. PS_1 will check the document whether it is modified or not, in no modification case the document is archived in PS_1 archive. Then U_1 is redirected to U_2's (Service Provider) web page.

11. Finally the document is mutually signed by both users, and now they are in possession of mutual signed document.

4.3.3 Advantages and Disadvantages

There are many advantages and disadvantages related to this scheme.

Advantages

- The document is downloaded to user's computer where he has control over the document in the mutual signing model
- User is choosing the security level for signature
- A compromised proxy signer's private key does not compromise the users' private key
- User is validating the signature on his computer which satisfies the requirements given by Difi
- User is actively involved in the signing process

Disadvantages

- Long computation time i.e. simple computer cannot tackle heavy cryptographic computations which could lead to slow validation process thus the system could be difficult to use and usability property will be breach. On the other hand increasing the key length to 2048 bits will increase the computation time even more
- Not supporting other types of document than PDF
- The user is dependent on proxy signer.

4.4 Proxy Signer Centric Proposed Solution

In this scheme proxy signers are working actively in signing a document. More over proxy signers are talking to each other and to the corresponding user i.e. PS_1 to U_1 and PS_2 to U_2 and so on. The central trust party of this model is eID provider which makes the scheme more trustworthy.

4.4.1 System Architecture

In this scheme asynchronous web services are also assumed since implementing synchronous web services is cumbersome in this case. In this scheme it is assumed that a document is meant to be signed mutually by both parties. U_2 is assumed to be service provider and U_1 is a service user. U_1 is not logged in to U_2's web. For the sake of simplicity of figure error handling are not considered here. The dialogue between U_1 and U_2 are carried out in advance. Figure 4.10 below illustrates high level system architecture.

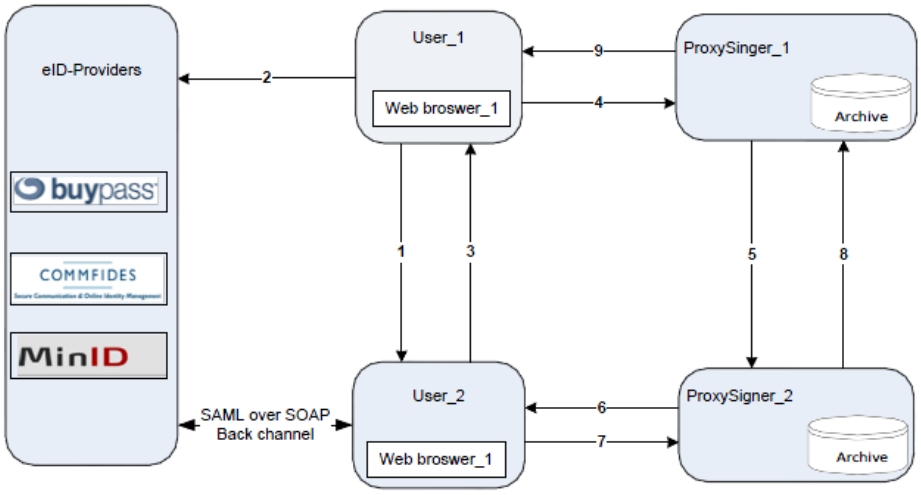

Figure 4.10: High level System Architecture of Proxy Centric signing solution

A document is created as result of dialogue between U_1 and U_2 where they are agreed to sign an agreement mutually. U_1 requests (1) U_2 to send the document for signing, before U_2 sends the document U_1 must authenticate. As in previous solution he will be directed (2) to the authentication page of U_2, where U_1 chooses an eID provider whom U_1 has association with. One of the approved eID provider will authenticate U_1. Authentication data will be sent directly to U_2 over a secure back channel (this step is server side action i.e. not visible for U_1), and thereafter U_2 will send the document to U_1 (3). U_1 will read the document and send it to his PS_1 and asking for signing on behalf of him (4).

The main difference between this solution and the previous solution comes here; when PS_1 received the document it will check the authenticity of U_1. If U_1 is authentic PS_1 signs and formats the document and sends it to PS_2 (5) (which is given by U_2 where to send the signed document), and a temporary copy will be saved in archive. PS_2 validates the document and presents it to U_2 (6). If U_2 agrees the content of the document he will send a signing order to PS_2 (7) to sign

on behalf of him. On receipt of signing order PS_2 will sign the document and send it back to PS_1 (8) and a copy will be saved in PS_2's archive for later use. PS_1 will validate the signature, and if valid present it to U_1 (9). The document will be compared with the temporary saved document in PS_1's archive, if they are identical the mutual signed document is saved in PS_1's archive and the other one will be discarded.

4.4.2 Process Flow

In this solution the proxy signers are acting more than users. The same technologies are used as in previous solution, described in section 5.3. The main difference between this solution and the previous solution is involving proxy signers more than the users. U_2 is assumed as a service provider which offers services to U_1. Mutual authentication is not necessary as U_2 i.e. service provider is authenticated through SSL certificate where certification authority (CA) guarantees the identity of U_2, but U_1 must be authenticated since assumed that U_1 is not logged in to the U_2's web page. Steps 1 through step 2 are the same, but in step 3 a slight change in the parameters to be sent to U_1 is occurred. Figure 4.11 illustrates the process flow. For the sake of simplicity of the figure error handling is not considered.

1. A dialog between U_1 and U_2 with an initiative from U_1 is carried out. U_1 requests for a service and U_2 requests U_1 to authenticate him. U_1 is directed to authentication page where U_2 trusts in (as in previous solution authentication of U_1 is necessary).

   Since U_1 is directed to authentication page U_1 chooses an eID provider (which could be Buypass, commfides or MinID) to authenticate. This could be done using an applet connected to eID provider.

2. Electronic ID (eID) provider will authenticate U_1. As mentioned in previous solution that this could be achieved using an applet connected to eID provider, but if MinID is used for authentication U_1 will again be redirected to ID-porten (MinID server) for authentication. The security level for authentication is 4 (personal high) for using an approved eID provider. Meanwhile all authentication information, included time stamp, will be recorded with eID provider according to the eID policy. Succession of authentication will also be acknowledged to U_1. Directing U_1 back to U_2's web page will indicate that the user is authenticated when MinID is utilized. Using eID provider with an applet solution, it will be indicated in the applet that the "authentication was successful".

3. When U_1 is authenticated U_2 will send document to U_1 for signing. This is achieved through directing U_1 to file archive where U_1 is granted limited access rights to this archive, and U_1 will download the document to his own

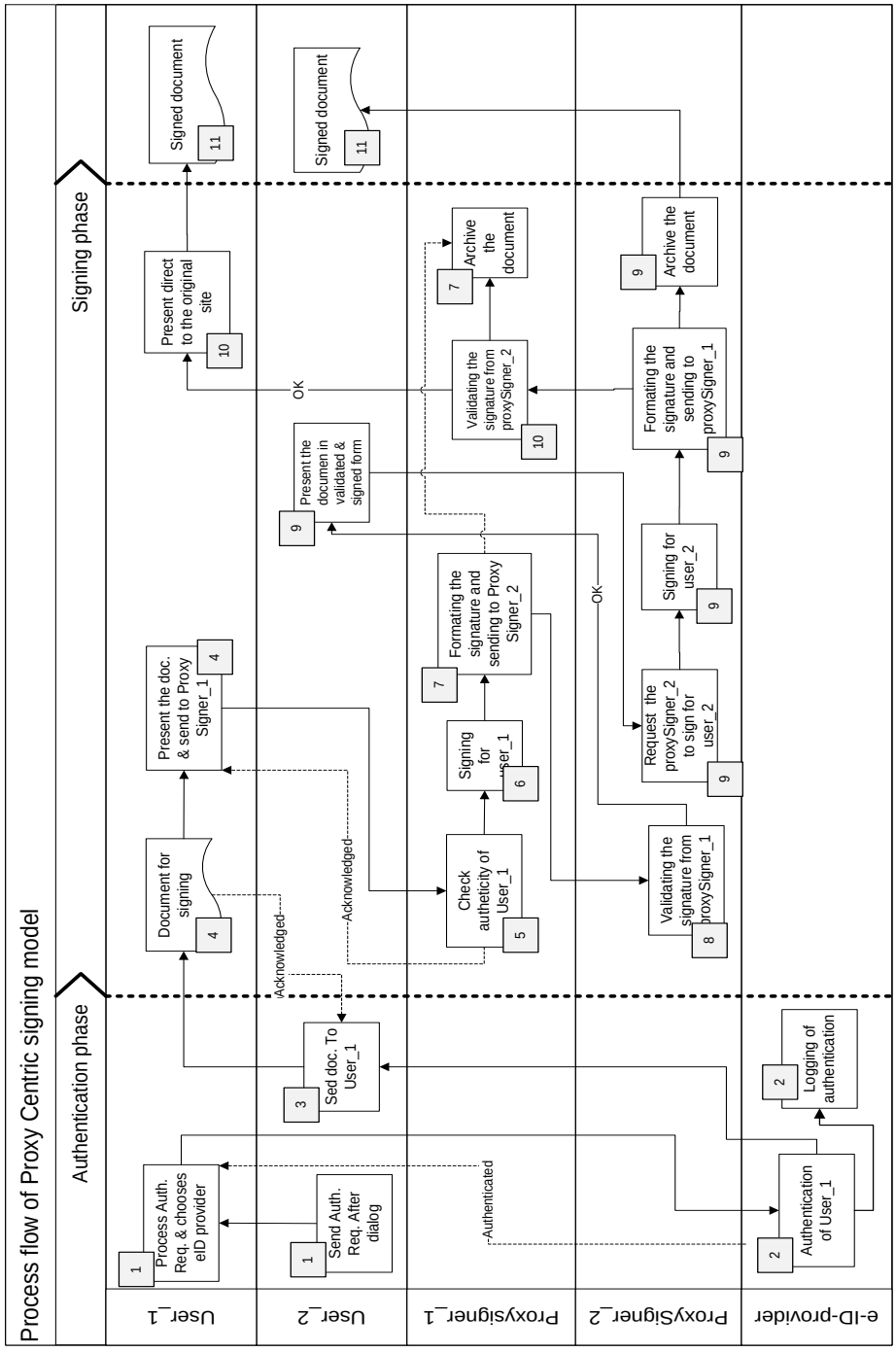

Figure 4.11: Process flow of Proxy Signer Centric signature solution

machine for further actions. In other words the document is in U_2's server where U_1 is given access through a secure channel to download it. Secure channel is provided using SSL/TLS certificate.

Along with the document the corresponding parameters will be sent to U_1 as well. These parameters includes personal information of U_2 i.e. service provider's organization number, name and id of the case worker who deliver the service and so on, URI where the singed document to be sent to (the URI of proxy signer U_2 uses for signing), a unique identifier of PS_2, unique identifier of the document, what security level is needed for signing this document, deadline for downloading the document (this could be few minutes otherwise timeout will kill session which can lead to requesting re-authenticating the User) and many more. A complete list of parameters is implementations dependent.

4. U_1 receive the document and acknowledge the receipt of document. Acknowledgement leads to killing the session. U_1 retrieve the document and read the content of the document. U_1 invokes PS_1's web services to send the document with the corresponding parameters. The parameters included are U_1's personal information, URI received from U_2 for where to send the signed document, unique identifier of PS_2 and unique identifier of the document. Along with these parameters, a signing order in cryptographic form as discussed in section 4.1.2 is sent to PS_1 which will enable PS_1 to sign the document. For transmission of data between U_1 and PS_1 SOAP over HTTP/S is used. This will secure the content of the document in the link between U_1 and PS_1 which achieve the confidentiality properties of the document. When PS_1 received the document it will acknowledge the receipt of document which also leads to decouple the SOAP session (asynchronous web services).

5. When the PS_1 receives the document it will first check the authenticity of U_1. This is achieved through checking the authentication data as discussed in section 4.1.2. The confidentiality is secured by SSL/TLS certificate between U_1 and PS_1.

6. PS_1 signs the document on behalf of U_1 which is described in detail in section 4.1.3 . How secure signature is created, is discussed in section 2.4.2 where some threats are also considered. Security issues which can occur during the key generation are also considered.

7. PS_1 formats the signature and prepares sending it to PS_2. In formatting PAdES-LTV standard is used where this standard has many advantages over for SEID-SDO which Altinn is using in the existing solution. A copy of the semi-signed document will be archived till the mutual signed document is received by the PS_1. Alternative a hash of the signed document will be computed and archived till the mutual signed document is received and validated. This is to prevent the modification of the document by U_2. The formatting of the signature will be discussed in more detail in section 4.1.3.

8. PS_1 uses unique ID of PS_2 to identify PS_2 and the URI (sent in step 4) to send to send the document to i.e. to PS_2. When PS_2 receives the signed document it will validate the signature received from PS_1. Before validating the signature it will extract the document and check the timestamp as well as the validity of certificates of Certification Authority (CA) certified the PS_1's certificate and authenticate PS_1, the certificate of the CA who certified the public key of U_1. This is achieved by checking the certificates using Online Certificate Status Protocol (OCSP) and Certificate Revocation List (CRL) invocations. If these certificates are valid then PS_2 will validate the signature which would be discussed in detail in section 4.1.3.

9. After the signature is validated PS_2 sends the document U_2 for approval i.e. the signature was valid but U_2 must read the document before he orders signing. In this step the document is presented to U_2 where U_2 will both approve the document and send a signing order to his proxy signer to sign the document on behalf of U_2 or reject the document. When PS_2 receives the signing order it will validate it as described in section 4.1.3 and sign the document. The same procedures carry out as in step 5, 6and 7 where after formatting the finally singed document is sent to PS_1. A copy is saved in PS_2's archive for the later user.

10. When finally signed document is received by PS_1 it will validate the signature as done by PS_2 in step 9 and it will also check whether the document is modified or not. This is achieved through computing a hash of that part computed previously (i.e. the content of the document and signature of U_1) and checking it with hash of previously computed and temporarily archived in step 7. PS_2 informs U_2 by sending a link to download the document from PS_2's file archive. Since they have pre-shared authentication data which could be used to download the document from PS_2's file archive.

    Alternative the document could be sent to U_1's email using secure email exchange mechanisms, but this would be quite challenging for a proxy signers servers to do.

    U_1 sends an approval message to PS_1 in order to archive the document.

11. Finally the mutually signed document is handed by both parties i.e. U_1 and U_2.

4.4.3 Advantages and Disadvantages

There are both advantages and disadvantages related to this proposed solution. In contrast with previous method this eases the computation which should be done by users.

Advantages

- With increasing the encryption key from 1024 bits to 2048 bits will also increase the computation time. Heavy cryptographic computation by simple computer is inconvenient. This method will ease the computation time by letting proxy signer to do it, because proxy signer has this capability.

- Using this method will protect users private key from compromising, because proxy signer will be issued private key extended from user's private key, as described in section 2.4, that original private key cannot be derived from extended private key. If proxy signer's private key is compromised the user will revoke the proxy certificate and will issue new private key. So the user's private key is protected despite of compromising proxy signer's credentials.

- Using single sign on can lead to compromising user's private key but using the proxy signer's concept will protect it.

- Using proxy certificate will ease implementation, since proxy signer uses the same format as X.509 PKI certificate.

- Using proxy certificate will limit the delegation to different proxy signers i.e. the delegation has no cascading path. In other words a proxy signer cannot delegate its signing capability to a new proxy signer, thus increasing strong forgery property of proxy signature.

Disadvantages

- According to Norwegian eSignature law the signing must be carried out by the tools which are in possession of signatory, this method will breach this property of the signature

- More server side's actions which could be inconvenient for the user

4.5 Signature between Private People

In a question of how to sign a document when two parties do not rely on each other nor have websites to carry out the document exchange as an interaction between users i.e. as in service provider and service user model, another model is proposed in order to overcome this situation. This proposal is based on the following scenario; a Norwegian citizen living in Stavanger wants to rent out his apartment in Tromsø. The owner of apartment (called landlord thereafter) posts the rental announcement on Facebook. A person is interested to rent (called tenant thereafter) the apartment begins to chat (assuming both are online) in order to come up with an agreement. They agree to do a deal where both must authenticate to an approved Mail Archive (like digipost or Altinn can also be used). Tenant gives important information to the landlord (assuming that he trusts giving just personal number and address). Landlord writes the contract and signing mechanism starts.

4.5.1 Sytem Architecture

Since neither party relies on each other then mutual authentication is necessary. A new entity called mailing archive is added to this system where both users can exchange the document through this mail archive. Error situation is not included in the model for the sake of simplicity. It is assumed that both users had a dialogue through a public channel facebook in advance. Based on scenario U_1 is tenant and U_2 is the landlord. Figure 4.12 illustrates high level system architecture.

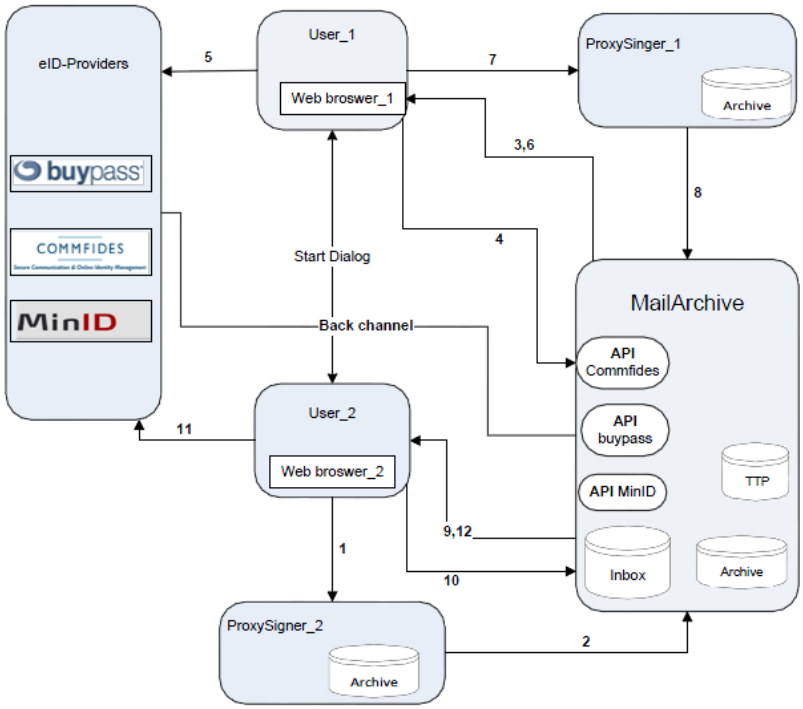

Figure 4.12: System Architecture of signing between private people

An agreement made between U_1 and U_2 where they have agreed that they will sign the document utilizing their proxy signers and they will authenticate themselves using eID providers. The document is sent to by U_2 to PS_2 (1) asking for signing where PS_2 will sign and send it further to Mail Archive (2). Mail Archive informs U_1 that a new post is received (3), U_1 tries to login (4) where he will be directed to eID provider web page to authenticate (5). Upon authentication succession U_1 downloads the document (6). U_1 reads the document and decides whether to sign or discard (not shown in the architecture), if signing he will send the document to PS_1 (7) and asking for signing on behalf of him. PS_1 signs the document and sends it to Mail Archive (8) where Mail Archive informs U_2 (9) of receipt post. U_2 tries to login (10) he will be directed to eID provider (11) for authentication. Electronic ID (eID) provider authenticates U_2, where U_2 can download the signed document (12).

All non-denial information saved in the TTP-archive of Mail Archive and a copy of document also saved in archive of Mail Archive. Users have rights of collecting non-denial information through Proxy Signers if needed. A copy of document is also stored in proxy signers' archive also.

4.5.2 Process Flow

In this case the following assumptions are made;

- Both users have eID which they can use to identify their selves.
- Both users have their own Proxy Signer for signing signature.
- Asynchronous web services are utilized.
- A dialog on a public channel has been carried out.
- U_1 tenant and U_2 is landlord in scenario.

For the sake of simplicity of the diagram error situation are not included but they could be handled in implementation. Figure 4.13 illustrates a detailed process flow based.

1. A dialog between U_1 and U_2 with an initiative from either U_1 or U_2 is carried out where U_1 is requesting for a service, for example signing a contract. In this case U_1 (landlord) initiates the process. This dialogue could take place on public chat room like msn messenger, skype, smartvoip, facebook and so on. In this case facebook is used. An agreement is accepted by both parties that they will authenticate themselves to each other using eID provider via Mail Archive.

2. U_2 sends the document to PS_2 and requests for signing and further sending to U_1's mail inbox in Mail Archive. U_2 invokes PS_2's web services for sending the document. As in other solutions SOAP over HTTPS is utilized for communication. Along with the document a set of parameters are sent to PS_2 such as U_2's personal and contact information, unique identifier of the document, personal number of U_1 in order to put the document in his mail inbox of Mail Archive. As described in section 4.1.4 that this Mail Archive uses personal number of U_1 in order to find contact information of user. In addition to parameters a signing order, as described in section 4.1.2, is also sent to PS_2. Confidentiality is achieved through SSL certificate.

3. When document is received by PS_2 it acknowledges the receipt of document which leads to decoupling SOAP session. Before signing the document PS_2 checks the authenticity of U_2 which is achieved using authentication data added in the header of the message which carries the document. Authenticating U_2 will approve that U_2 delegated his signing capability to proxy PS_2. Creation and validation of authentication data is described in section 4.1.2.

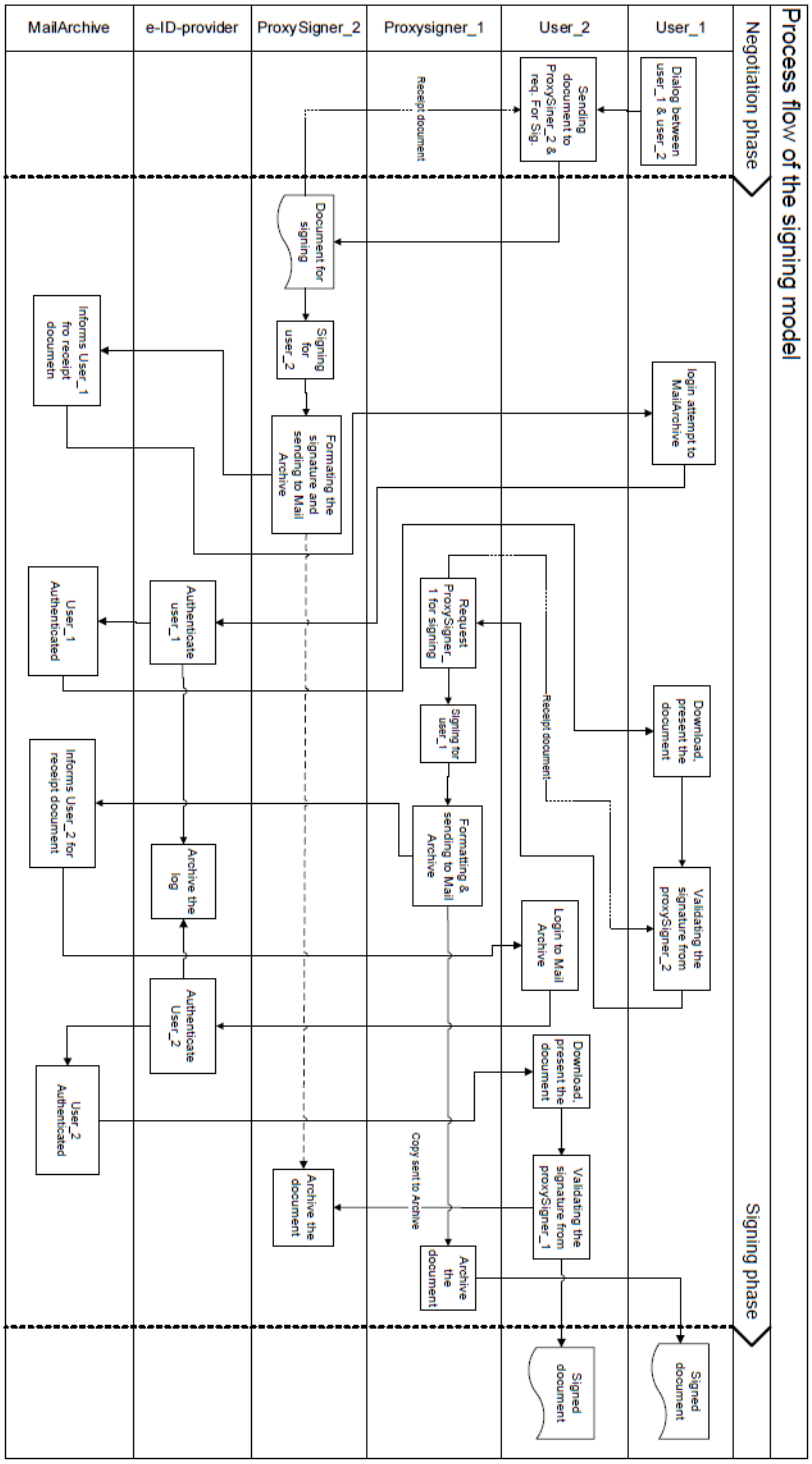

Figure 4.13: Process Flow of the signing between private people

When U_2 is authenticated, PS_2 will sign the document on behalf him which is discussed in section 4.1.3.

4. PS_2 formats the signature, as discussed in section 4.1.3, and sends it to the U_1's mail inbox in Mail Archive. IETF standard, PAdES-LTV, is utilized for the formatting. A copy of the semi signed document is saved in order to prevent modification by both the outsider (attackers) and U_1. PS_2 invokes Mail Archive's web services (SOAP over HTTPS) to send the document to Mail Archive.

   Along with the document parameters, necessary for further treating the document are sent. Unique identification of document, email address of U_1 (in case U_1 has not used mail archive previously), personal number of U_1, the title of the document and security parameters. Other parameters, which could be relevant and appear to be necessary during the implementation, will also be sent.

5. Mail Archive receives the document and acknowledges the receipt of document. This leads to killing the SOAP session (asynchronous web services).

6. U_1 must login to Mail Archive to download the document. Login demands authentication using eID provider. An applet solution will be used to authenticate U_1 to Mail Archive. Mail archive stores a copy of document in its archive and all logging information of U_1 is also saved in TTP.

7. When U_1 is authenticated the document is sent to U_1 and he downloads the document from Mail Archive to his computer and first validates it. Validation is discussed in detail in section 4.1.3. Upon acceptance of the document's content and succeed validation U_1 will invokes web services (again SOAP over HTTPS will be utilized) of PS_1 to send the document the document to PS_1 to whom he delegated his signing capability.

   Along with the document the corresponding parameters are also sent to PS_1. Which parameters should be sent depends on what Mail Archive one uses; if Altinn is used there would be some parameters while using for instance Digipost other information would be used.

   Mandatory parameters are to be sent in either cases are as follows; Unique id of document, personal number of U_2, the title of the document, security parameters as discussed in section 4.1.3 and other parameters which could be relevant and appear to be necessary during the implementation.

8. When PS_1 receives the document it acknowledges the receipt of document which leads to decoupling SOAP session. PS_1 authenticates U_1 whether he authentic and has delegated his signing capability to PS_1. This is achieved using authentication data received from U_1. Creation and validation of authentication data is described in section 4.1.2 and section 4.1.3 respectively.

   When U_1 is authenticated, PS_1 signs the document and formats the signature as discussed in section 4.1.3. The signed document is sent to Mail Archive by

invoking Mail Archive's web services and using the personal number of the U_2 as a unique identifier of U_2, where Mail Archive will put the document in U_2's mail box.

Along with the document, parameters necessary for further treating of the document are sent. Unique id of document, email address of U_2, personal number of U_2, the title of the document, security parameters as discussed in section 4.1.3 are among the important parameters. Other parameters which could be relevant and appear to be necessary during the implementation will also be sent. The authenticity and confidentiality between PS_1 and Mail Archive is achieved using SSL/TLS. Proxy Signers and Mail Archive must have security association in order to sent document to each other in safe manner.

9. In the same way as in step 5, Mail Archive informs U_2 of receipt document. This is achieved through sending either sms or mail to an email address U_2 registered in Mail Archive. U_2 must authenticate to Mail Archive in order to download the document.

10. U_2 attempting login to Mail Archive which would redirect U_2 to its authentication page. By choosing an eID provider an applet will appear letting U_2 to authenticate. Upon succession of authentication U_2 downloads the document from his inbox in Mail Archive. The confidentiality is achieved using SSL/TLS between U_2 and Mail Archive. Mail Archive saves a copy of document in its archive and log all logging information (not shown in the figure) i.e. non-denial information in TTP for later use.

11. When U_2 downloaded the document it starts validating before accepting it. He validates the document and reads the content of it. In case of acceptance i.e. no modification or altering the document on the way or intentionally by U_1 is appeared he sends a copy to PS_2's archive as a proof for later use. This replaces the semi signed document saved in PS_2's archive in step4.

12. The mutual signed document is received by both parties.

Document is signed mutually by both users and archived in their Proxy Signers' archive, and downloaded in electronic form to their computers as well. A copy is also saved in archive of Mail Archive, as well as non-denial information is logged in TTP. This overcomes the problem of signing mutually by private people who are not officially service provider and neither have web sites in order to achieve it as in other solutions.

4.5.3 Advantages and Disadvantages

There are both advantages and disadvantages with this solution proposal, and they are described below.

Advantages

Most of the advantages of this solution are common with the previous solution, but some of them are more specific to a single solution. Only specific advantages will be presented here.

- One of the specific advantages is that this solution do not need any web side in order to carry out signature, and exchange of the document, as an interaction in the service user and service provider model.
- Ease of implementation; since user needs to integrate the signing application with proxy signer and the proxy signer with mail archive. Integration of eID with mail archive already exists.

Disadvantages

- Dependent on several entities in the system such as Mail Archive, proxy signers, eID provider
- Poor privacy at the beginning, how ever they do not rely on each other, but yet giving the personal information to U_2 (in this case).
- The user must rely on mail archive which could be a weakness of the system.

Chapter 5

Analysis

This chapter presents an analysis of the proposed solutions. First some requirements are derived in order to understand what the system should perform, and then all these requirements are analyzed using use cases. All technologies and standards meant to be utilized are also analyzed. Vulnerabilities of each technology are described in more detail.

5.1 Requirements for the Proposed Solutions

Requirements describe features of a desirable system. In order to understand better how the system should work and what conditions the system should satisfy as a whole system, some requirements are derived. These specifications are supported by technical reports, standards and relevant book [27] studied in during the work. All requirements are common for all three solutions except 1st requirement, where in the third solution the system must authenticate the user to Mail Archive not to service provider.

The requirements specifications include functional, non-functional and not least legal.

5.1.1 Functional Requirements

Functional requirements define the features a system must perform [27]. The system;

1. Must authenticate a user to a service provider using eID provider.
2. Must carry out a signature and verify it.
3. Must be able to check certificates validity embedded in the document.
4. Must protect confidentiality and integrity of document.

5. Must support long term validity of a signed document.

6. Must identify the signatory and ensuring non-repudiation of a document.

5.1.2 Non-Functional Requirements

According to [27] Non-functional requirements define features relevant to the system performance.

- User Friendly, Usability: The system must be user friendly i.e. easy to understand and easy to use.
- Availability: The system must be available all the time when it is required to be utilized.
- Reliability: The system must be reliable i.e. the expectation of error occurring must not exceed threshold.
- Cost Effective: The system must be cost effective in order to implement it.
- Speed: the system must be as fast as the user not to be bored of using the system.

5.1.3 Legal Requirements

In every country technology's evovelmaent is regulated by law, where the authorities set limitations and requirements to be fulfilled when a technology is to be implemented. In Norway, Department of Justice in association with Agency for Publi Management and eGovernment (Difi) and the Post and Telecom Authority regulate evolvement and implementation of these technologies. The legal aspects are considered by Department of Justice while development and implementations are regulated by Difi, and misuse prevention is regulated by Post and Telecom Authority. Two legal requirements are considered here.

- The signature must be in a state to be accepted as a hand written signature
- The signature must satisfy the definition of eSignature law

5.2 Requirements Analysis

In order to capture the functional requirements of the system some use cases are developed, and based on these use cases the system's functionality is validated. The motivation of using use cases is to prove that the system works as intended. Non-functional requirements could be analyzed when the system is implemented because there are some moments that should be evaluated by users of the system. Legal requirements are approved through Norwegian law. The functional requirements analysis is desirable of this thesis.

5.2.1 Use Case Diagram

Use case diagram is a formal way of analyzing the system. This method is developed by IBM Research [34]. The main objective of use case diagram is to capture a feature of a system. According to [83] uses case diagram plays an important role in mapping the functionalities of a system in a way to be understandable for users, developers and analysts. It means that use case diagram provides a common understanding of the system by all parties involved in the developing process.

Although this system is not implemented but use case diagrams are developed in order to view the main objective of the system. Since the system focuses on signing, where authentication should be carried out in advance, in order to trust the signature is performed by a valid person.

Use Case 1
Authentication of $User_1$ to $User_2$ (Service Provider)

This Use Case diagram will cover the following functional requirements; (1) the system must authenticate U_1 to U_2, (6) the system identify the signatory. This use case is common for all three solution proposals. The description of this use case is given in table 5.1. Figure 5.1 illustrates the Use Case diagram. All tables can be found at the end of chapter 5.

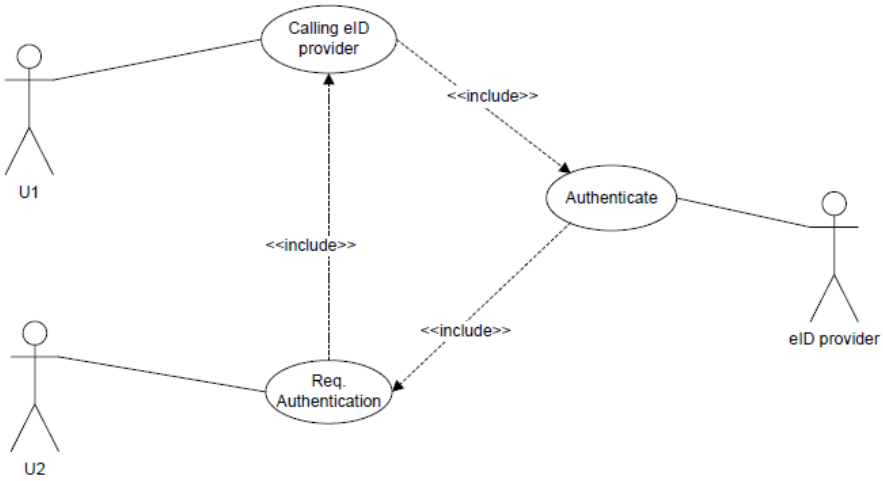

Figure 5.1: Use Case 1, Authenticting U_1 to U_2

Figure Use Case 1, Authenticting U_1 to U_2

Use Case 2
Singing Document and Verifying the Signature

This Use Case diagram will cover the following requirements; 2-5. In this diagram a high level service description is presented, and figure 5.2 illustrates it. The specific use case diagram for capturing the signing and formatting functionality is presented later.
Table 5.2 describe the textual description of Use Case 2.

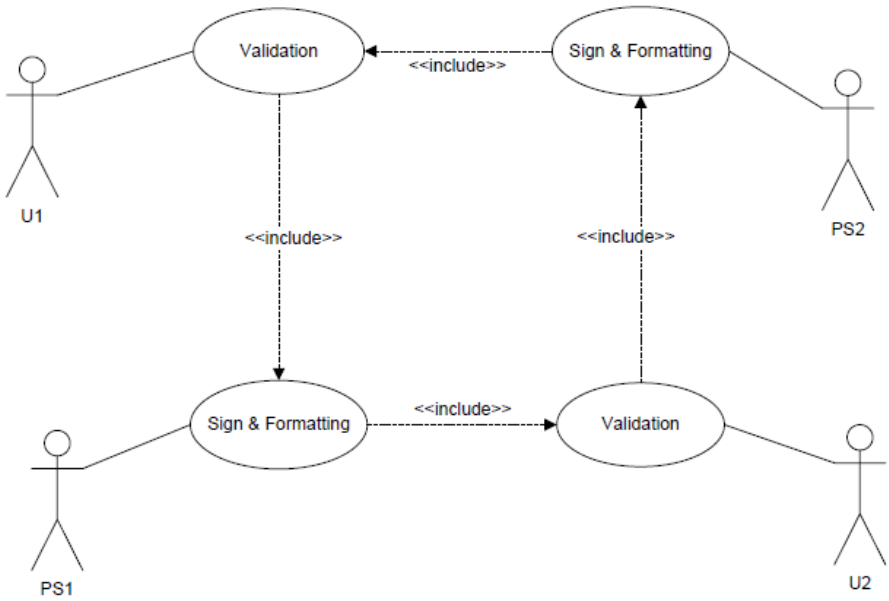

Figure 5.2: Use Case 2 (Signing and Formatting the document)

Singing Document and Verifying the Signature

Use Case 3, Signing and Formatting

Use case 2.1 , 2.2 and 2.3 which start at step 2 and 3 of Use Case 2, where Table 5.3, Table 5.4 and Table 5.5 show the description of them respectively. Figure 5.3 shows the use case diagram. In the first step of 2.2 another use case 2.2.1 applies where the textual description is presented in table 5.6

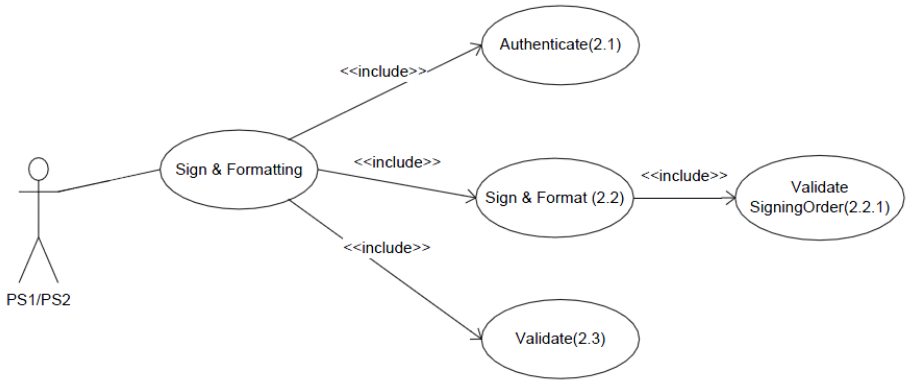

Figure 5.3: Use Case 3

5.3 Technologies and Standards Analysis

In this section technologies to be used in these proposed solutions are analyzed, where their advantages and disadvantages discussed. This analysis will answer the question of why these technologies are preferred in spite of many other technologies which could be used. Since entities in all proposed solutions are the same, then similar technologies will be utilized.

PKI

The most important technology which supports security and trust of the system is public key infrastructure (PKI). One of the important function of this technologies is to provide and manage trust relationship between entities connected through Public Key Infrastructure (PKI). This technology is described in section 2.1 in detail.

Many other technologies could be used to implement these solutions but why PKI? There are many reasons for why PKI is preferred.

PKI according to [76] has strong security than other technologies. First of all this technology uses asymmetric encryption which is of desire, because of strengths and initiation in a business deal online without using PKI is impossible. An example of this could be a user agent wants a service from a service provider where the do not know each other previously. So, it is obviously that service provider do not rely on user agent. In order to make a deal they must sign an agreement, but how to initiate signing. PKI is a possible solution where pre-shared security parameters are not necessary. Initiation of a deal will be started by service provider, which offer a service and requests user agent to authenticate before user agent can serve

the service. Service provider will use SSL/TLS to authenticate to user agent.

Signing such an agreement using symmetric encryption mechanism is demanding beforehand exchanging of security parameters for example pre-shared encryption and decryption keys. This is impossible in case a service requested first time by a user agent in e-commerce, but PKI solve this problem because no pre-shared security association is needed.

As mentioned earlier that no stronger and better technologies than PKI found today, therefore PKI is chosen to use.

Electronic ID (eID)

Electronic ID provider is the central trust party that these systems rely on. Electronic IDs are issued by approved eID providers. According to Post and Telecom Authority [71] all electronic ID issuer must be registered with Post and Telecom Authority. In Norway there are three eID providers, namely buypass, Commfides, and the third eID provider is a national authentication portal called MinID where everybody can use it freely.

Electronic ID i.e. digital certificate which could be used to produce digital signature, is trustworthy and secure in their use [76]. Therefore digital signature must replace handwritten signature which has been reliable for hundreds of years. Electronic ID is microprocessor-based smart card which is certified by approved Certification Authority (CA) which is again based on PKI.

In eID card lots of information needed for authentication, encryption and decryption are stored. Among them private key, public key are used in connection with encryption and decryption.

According to [74] RSA with key length of 768 bit is broken and it is assumed that a key length of 1024 bit might be broken within 2015. It means eID are (smart card) is vulnerable within 4 years. Increasing the key length to 1536 bit or 2048 bit is a solution to overcome this problem and convince the users, but this has a disadvantage of increasing of computation time. The former could be better now because increasing the key length from 1024 bit to 1546 will increase the security dramatically but will not increase computation time such that encryption and decryption time become inconvenient. As technologies evolve very fast where RAM getting larger and processors getting faster, which will overcome the computation time problem, but rather fasten breaking key process and decreasing key braking computation time.

Key length of smart cards today is 1024 bits which is meant to be secure within 2015. Buypass also uses 1024 bit key in the eID card, but they are considering increasing the key length [10], Although the link shows that they will increase till the

end of 2010. This means that the key must be updated, but the security manager, I had contac with last time, said that they have not updated yet.

In eID micro-processor-based smart card is utilized. This card has an operating system in order to compute different data of desired. MULTOS is an operating system utilized in smart card [32], [14]. This operating system is also used in buypass smartcard [11].

According to [33] the most important advantages of smart cards are as follows;

- Capable of encryption and decryption
- With a high level of security can implement both parties requirements
- Secure against counterfeits attack for ex. if a smart card is missed
- Smart cards can proceed independently from a back end system or offline

Since keys stored in smart card are very sensitive then this storage must be secure. According to [37] smart cards important functionality is to provide secure storage. Acccording to [41], [**?**] an alternative solution to store the keys in secure manner, is to compute the secret keys on base of smart card's serial number and store them in ROM, but this will be extra computation for microprocessor when the keys should be retrieved. Microprocessor play important role in smart card which required being secure against physical and side-channel attack [1].

The reasons for choosing smart card in eID is security since microprocessor-based smart card (used in eID) makes cryptographically protected communication possible [80].

Proxy Signer

The concept of proxy signer is first studied by Mambo et. al.. An original signer delegates his signing capability to a proxy signer (a third person). In this project it is meant to be a server performing Proxy Signer's role.

There are many types of open source software to be developed as a proxy signer server. OpenSource PDF signer server which could be used to develop a signing server, based on proxy signer concept. The advantage of using proxy signer as signing server is that carrying out heavy cryptographic operations on a simple computer are cumbersome and inconvenient. Delegating this operation to a server with the ability of signing is a good idea. Sevral studies yielded that the proxy signer concept is strong engough for delegating signing capability to a third party.

Security Assertion Markup Language (SAML)

As described in [58] that SAML is a XML framework for creating, requesting and exchanging security assertions between the entities. This standard is intended to be used in proposed solutions because this standard is extensible which makes it possible for further development [73]. OpenSource Shiboleth, SAMLphp are based on SAML.

When a user authenticates through eID provider, eID provider must send the authentication data directly to service provider. This data (including parameters) are sent through a back channel. In this case SMAL according to [56] is a good solution to achieve it. This means that when user sends parameters (authentication data) to eID provider, where eID provider validates and sends SAML token containing security information directly to service provider on a secure channel.

As described in [56] that SAML carries authentication, attributes and authorization information asserted in the SMAL message, where all information is encapsulated in the SOAP over HTTP protocol. The SAML request or respond is encapsulated in the body of SOAP message, where the SOAP message is further embedded in body of HTTP message. Figure 5.4 illustrates it.

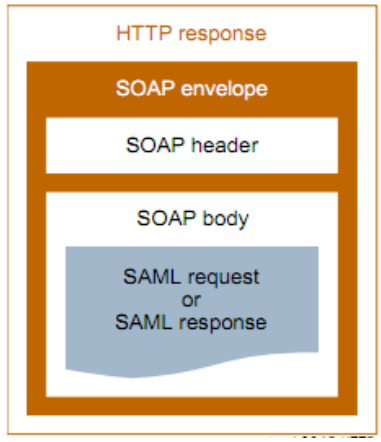

Figure 5.4: Encapsulation of SAML messeage in SOAP [56]

So data and parameters from eID provider are embedded in the SAML message where SMAL message is embedded in SOAP message, and it is further encapsulated in body of HTTP message.

The vulnerabilities of this standard are mentioned in section 3.4 and will not be repeated here. But as described, message replay and man in the middle attacks are overcome. The remaining attack i.e. attack by information leakage according to [60] can be covered by using both SAML with SSL/TLS. It means the authenticity, confidentiality is covered by SSL/TLS and the integrity (and confidentiality) is

covered by SAML. Confidentiality will be secured twice i.e. sending authentication data and necessary parameters directly from eID provider to service provider are secured twice. In SAML V2.0 most of attacks are overcome [60].

Since SSO is desirable, which could be achieved easily by SAML, and sending SAML token directly from eID provider to service provider is important, which could also be acheieved by SAML, are the main reason for using SAML standard in the proposed solutions.

Web services

Web services are described in section 2.2. There are several advantages with web services, among them few are discussed here.

Interoperability is one of the most important feature where developer are given opportunity to develop services with their preferred programming language i.e. web services is platform independent [6]. In addition it offers non-proprietary route to their solution because it works outside of a private network.

Reusability of components increases development efficiency. Since Difi assumed to use asynchronous web services but designing asynchronous web services is challenging [61]. This property makes an application cost effective if it is achieved in best way.

Deployability, web services have this property which make it possible to deploy web services over a fire wall to servers [38].

Since web service is based on XML then Advantages (simple, readable, widely supported and so on) and disadvantages (verbose message, ease of use means a proliferation of a standard) of web services are also based on XML. It means advantages of XML are advantages of web services and so disadvantages [17]. So disadvantages of web service are verbose i.e. human readable and ease of means of a proliferation standards. These disadvantages are not fatal in order make problem for implementation and security of a system.

But a major problem in web services is the usability problem with web services. Usability is important for the end users, because end users utilize an application, then it must be taken care of. An application with low usability will not be desirable for the end user. NIST in [70] states that there are still many problems with usability of web services where Difi did not mention how to solve them neither.

Despite of usability problems and disadvantages mentioned above, advantages of web services are more accurate. Web services make it easier for the developer to develop and implement the application. Taking all these advantages in consideration, web services are intended to be used in proposed solutions.

SOAP

Some applications communicate together using Remote Procedure Call (RPC) method for instance communication between objects like DCOM and COBRA. This method cannot be used for HTTP because RPC introduces security and compatibility problem with this method, for instance firewall will block this kind of traffic [81]. To overcome this problem Simple Object Access Protocol is developed.

SOAP provides the facility for the application to communicate with each other independent of underlying technologies and programming language.

Since communication on internet using HTTP and HTTP was not compatible with RPC, then SOAP was necessary [81]. This standard is widely used in web service [70].

In these proposed solutions almost all communication between entities are utilizing SOAP over HTTP/S. This improves the compatibility and security of communication channels.

Since SOAP is based on XML language and the XML standard is vulnerable to XML based attacks, then SOAP is also vulnerable to XML based attacks. Describing these attacks is out of scope of this thesis but to overcome this problem the security administrator of the system must deal with the issue at the application layer, which is not described in detail here. A possible solution according to [65] could be using XML-aware firewall which provides multi-layered security.

PAdES-LTV

In order to send a document over a communication channel it must be formatted in the way that the document is not altered and easily reach by the other party.

PAdES-LTV is an ISO 32000 extension where long term validation feature is added to the PDF document.

Utilizing PAdES-LTV in the proposed solutions was of desire of three reasons;

- Long term validation i.e. supports validation up to developers' choice [25] and as described in section 2.5.3 validation data can be further protected. This property is desirable for Lånekassen.
- International standard (not specific like SEID-SDO which is Norwegian standard)
- In PAdES the document content can be read after signature, which desirable, but in SEID-SDO the content of document cannot be read once the object is signed.

Long term validation in SEID-SDO depends on the lifetime of cryptographic parameters [68]. On the other hand in PAdES-LTV the developer has the opportunity to extend the validation lifetime by appending a new LTV structure validation data and Document Time-Stamp in order to extend the validity of the signature [53].

The advantages of this standard are mentioned above, but there is also disadvantage using this standard, for instance all types of document are not supported unlike SEID-SDO where three different types document like XML-based, CMS-based and PDF, are supported. According to [19] principals developed by Difi state that the system could be able to be extended such that a verification long country's boarder could be performed. In this case a SEID-SDO will not satisfy this condition but PAdES-LTV will do, because PAdES-LTV is an internation standard

Use Case 1	**Authentication of U_1 to U_2 (service provider)**
Description	U_1 authenticates to U_2 using eID smart card connected to eID-provider.
Actors	U_1, U_2 and eID provider
Assumptions	1. U_2 has association to eID provider, and U_2 is autheniticated through SSL certificate. 2. It is assumed that all communication channels are secure and no sniffing can occur during the authentication. 3. The dialog is carried between U_1 and U_2 in advance. 4. Assuming that the smart card is inserted in card reader.
Steps	1. U_2 is requesting U_1 to authenticate using an approved and trusted eID provider. 2. U_1 clicking on an eID solution which he has association with. This leads to calling eID provider. 3. eID provider asks for inserting eID card in card reader and enters PIN code. 4. U_1 enters his personal identity number (PIN) code of smart-card. By entering PIN code the smartcard will compute a login access signature which will be sent to authentication server of eID provider. 5. eID provider's authentication service will determine whether to authenticate or not. 6. Upon authentication succession the authentication data will be sent to U_2 in a back channel.
Variations	U_1 enters wrong PIN code

Table 5.1: Use Case 1, Authenticaion of U_1 to U_2

Use Case 2	**Signing, formatting and verifying the signature**
Description	U_1 wants to sign the document using PS_1 and sends it to U_2, and U_2 wants to sign the same document using PS_2 and sends it back to U_1.
Actors	U_1, U_2(service provider), PS_1 and PS_2
Assumptions	1. Users have derived private keys and proxy certificates to their representatives, proxy signers, to sign and verify signature respectively. 2. Authentication data and signing orders are already created by users 3. The channels between users and their proxy signers are secured using TLS. 4. Proxy signers authenticate theirselves using SSL certificates. 5. All channels between users and proxy signers are secured by using TLS. 6. Document to be signed has PDF format
Steps	1. U_1 inserts all necessary parameters to signature application and sends it PS_1 for signing on behalf of him. 2. PS_1 authentictes U_2 before sign the document(use case 2.1 starts here). 3. PS_1 signs and formats the document (use cases 2.1 and 2.2 start here) and sends it to U_2. 4. U_2 validates the signature (use case 2.3 starts here). 5. U_2 calls signature application, and then adds necessary parameters, i.e. authentication data and signing order, and sends it to PS_2 for signing on behalf of him. 6. PS_2 authenticates U_2, and signs and formats the document (use case 2.2 applies here again) and then sends it to U_1 7. U_1 validates the document (use case 2.3 applies here again).
Variations	1. Users insert one or more wrong parameters. 2. Certificate(s) embedded in the document is (are) invalid.

Table 5.2: Description of USe Case 2

Use Case 2.1	**Authenticating User to Proxy Signer**
Description	User wants to authenticate to proxy signer
Actors	Proxy signer server and User
Assumptions	1. Authentication data is previously created and is correct. 2. Relevant pre-shared keys and data, like private key for proxy signerare exchanged.
Steps	1. User sends authentication data and parameters to proxy signer. 2. Proxy signer collects its private key, issued by corresponding user, and gets the warrant message sent by user to concatenate them 3. Proxy signer computes hash of the warrant message concatenated with proxy signer's private key. Proxy signer extracts the data and gets the hash received from user 4. Proxy singer compares the computed hash with the received hash and decides to accept the user as a valid user, i.e. user who delegated his signature capability to proxy signer, or rejects him.
Variations	Wrong parameters are received by proxy signer
Issues	

Table 5.3: The table shows description of Use Case 2.1

Use Case 2.2	**Signing and formatting the document**
Description	Proxy signer signs the document
Actors	Proxy signer
Assumptions	1. Signing order received by proxy signer is correct. 2. User is authenticated to proxy signer. 3. Proxy certificate issued by the user is still valid
Steps	1. Proxy signer gets document with the signing order (use case 2.2.1 starts here) 2. Proxy signer concatenates the document with signing order and computes hash of them 3. Proxy signer collects the private key of the corresponding user. 4. Proxy signer signs the computed hash. 5. Proxy signer adds the signature to the document. 6. Proxy signer adds all certificates, OCSP response information, CRLs and miscellaneous parameters to DSS. 7. Proxy signer appends DSS to the document. 8. Proxy signer gets system timestamp. 9. Proxy signer adds timestamp to the timestamp document. 10. Proxy signer appends the document timestamp to the document.
Variations	
Issues	

Table 5.4: Use Case 2.2

Use Case 2.3	**Validation of signature**
Description	Pproxy signer/User wants to validates the signature
Actors	Proxy signer/User
Assumptions	1. All validation data is received by the user or proxy signer. 2. All certificate connected to the signature are valid
Steps	1. Proxy signer/user checks the timestamp against the system timestamp. 2. Proxy signer/user checks the validity of the certificates 3. Proxy signer/user gets signing order. 4. Proxy signer/user concatenates the document with the signing order and computes SHA1 digest of it. 5. Proxy signer/user retrieve the signature and decrypts it using the sender's proxy signer's public key. 6. Proxy signer/user compares the computed hash with the received (first decrypted) hash. 7. Proxy signer/user decides whether the signature is valid or not.
Variations	1. Invalid timestamp 2. Invalid certificate(s) 3. Invalid signature
Issues	

Table 5.5: Textual description of Use Case 2.3

Use Case 2.2.1	**Validating signing order**
Description	Proxy signer validating the signing order sent by the user
Assumptions	1. Signing order is correctly created 2. Warrant message is not modified
Actors	Proxy Signer
Steps	1. Proxy signer received the signing order along with the document. 2. Proxy signer gets the users public key received with the document. 3. Proxy signer decrypts the signing order and retrieves the warrant message. 4. Proxy signer concatenates the proxy signer's public key with the warrant message and computes a SHA1 digest of it. 5. Proxy signer compare the computed hash with the received (decrypted first) hash and decide for accepting or rejecting the order.
Variations	

Table 5.6: Description of Use Case 2.2.1, Validating Signing Order

Chapter 6

Validation and Discussion

This chapter presents validation, discussion. In validation the functionalities of all proposed solutions are validated against functional requirements. In addition the proposed solutions are also validated against the principals, for developing the common signature solution, given by Difi. Discussion will state the arguments that support those solution proposals will be an improvement of existing solution and Difi's solution proposals.

6.1 Validation

Functionality of all proposed solutions presented in this thesis are validated against functional requirements and by comparing the solutions to the relevant findings presented in the literature. Some of the non-functional requirements are also included. The comparing literature will be specified as the validation is made.

6.1.1 Validation of Functionality against Requirements

Functional requirements listed in section 5.1.1 are validated as follows;

Authentication U_1 to U_2 (Service Provider)

As seen in two first solutions that U_2 requests U_1 to authenticate, and U_1 is directed to $U_2's$ authentication page. For sending U_1 to $U_2's$ authentication page the redirect method will be used. U_1 utilizes one of the eID provider and authenticates to U_2. The authentication data will be sent directly to U_2 and U_1 will be redirected back to $U_2's$ web site. According to the technologies and standards presented in section 2, implementing these service are not impossible, though authentication of U_1 to U_2 is also possible.

Performing the Signature and Verification of it

The system carries out both signature and verification. As seen in section 4.1 both users and proxy signers are capable of verification. For signing, the system uses the PKC which is the most strong encryption solution in today's technology [76]. For verification the system adds public key certificate of users and proxy certificate of proxy signers to DSS of the PDF document. Since proxy certificate is extended from user's public key certificate, then signature can be validated by both certificates. Adding proxy certificate is adequate, then why adding user's public key certificate? The main reason is that if proxy certificate is revoked then the verifier has at least another certificate, original signer's certificate, to verify the signature. Actually the signature is connected to original signer (users) so users' public key certificate must be added to the DSS in the PDF document. As argued above that solution proposals can carry out signature and verify a signature, and thus the requirement is satified.

Validation of certificates embedded in the document

As presented previously that there are two mechanisms for checking the status of the certificates, thus Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP). The former method is used where the system can tolerate time delay and latter is utilized when time delay is not acceptable. The system uses these methods to validate certificates. Since these mechanisms are widely used in enterprises, and work without any shortcoming, it should also be working in proposed solutions. The above arguments are adequate for validating this functionality.

Protecting Confidentiality and Integrity of Document

The integrity of the document is achieved through signature embedded in the document. Formatting standards is utilized for protecting the signature, thus protecting the integrity of the document. Confidentiality is achieved through SSL/TLS mechanisms. These mechanisms are widely used in today's web technologies where security is also a goal. For instance, in bank industries SSL/TLS security is also used in normal web browsing. When entering to Internet banking it is obviously that the confidentiality of data exchanged between user and bank is encrypted. Proposed solutions provides both integrity and confidentiality of the document, thus this requirement is achieved and validated.

Protecting Long Term Validation

Long term validity is desirable of this project. This thesis's solution proposals provide long term validity of the signed document. This is achieved by using the ETSI standard ETSI TS 102 778 which is extended from ISO 32000 standard [25]. These standards are well known and widely used in the Internet. Based on ETSI standard, the system will provide long term validation of the document, thus this

functionality is approved.

Identifying the Signatory

Identifying the signatory is mandatory according to the definition of advanced electronic signature. PKI made it possible to achieve this goal of thesis. Public key certificate, certified by CA and embedded in the document, binds the owner's, i.e. the signatory, identity to the certificate [76]. Using PKI, thus public key certificate in signature, links the certificate to the owner (signatory), and thereby connect the signature to the signatory, will validate this requirement.

Usability Speed

As mentioned previously that in software developing process if the security is a priority, then usability will be weakened. Although usability is a non-functional requirement, it is considered in the proposed solutions. It seems that usability in these solutions will be weak. Since heavy cryptographic operations will be carry out during the signing, then the speed will be slow, and thus this will cause a poor usability. The usability will be as good as it can be validated, so usability in this level will be accepted as adequate.

Availability Reliability

The availability means that the system must be available all the time. This requirement can be validated adding redundant entities such that if one is down the other do the job. It is believed that the availability will be achieved as desired. Reliability means that the expectation of error occurring must not exceed the threshold. Redundancy will improve the reliability of the signature system. It is assumed that both reliability and availability requirements could be handled well, if the system could have been implemented. It is further assumed that the system will satisfy the minimum requirements of reliability and availability. Thus, these requirements are assumed to be validated.

Cost Effectiveness

Cost effective means that deployment and implementation of a system is cheap and easy. As seen that most of the existing technologies and mechanisms are utilized and therefore the system deployment and implementation will not be difficult. For instance, in 3rd solution proposal Mail archive is already existed. Therefore this requirement is also validated.

6.1.2 Validating against Difi's Principles

Difi has proposed some principals to be considered if common signature solution in the public sector to be developed[19]. These principals are briefly reviewed, and then proposed solutions are validated against them.

Principals

Different degrees of low / high coordination and standardization of the signature process provides superior set of 4 different possible images for an operational model, where each possibility will have uniqueness and special characteristics. The operation models are as follows;

- Diversification: low degree of coordination and standardization of signings processes.
- Coordination: a high degree of coordination and standardization of signings processes.
- Replication: the low level of coordination and high standardization of signings process.
- To make unique (Unifisering): high coordination and high standardization of the signing process.

The principals are divided in to three categories; superior principals, signature principals and verification principals, and are as follows;

Superior principals

1. Overall ICT architectural principles for the public sector to be followed.
2. Solutions for both signing and verifying must be arranged.

Principles aimed at signing

3. Signature will be accepted if eID-provider, who issued eID card, used for signing, has a contract with Difi.
4. Standardized formats available for signing.
5. Common solutions must cover basic needs for signing.
6. Service provider remains responsible for consideration.
7. Unified user interface for signing

Principles aimed at verification

8. Verification adapted for self declared eID-providers.

9. Verification Service as a common component.
10. International verification possible in the long term.

The validations of proposed solutions of this thesis against the principals are as follows;

Principal 1

As seen that the 1st principal is validated since ICT architectural principals are followed. Because the existing signature system performs the same operations, which they have followed the ICT architectural principals, as the proposed solutions of this thesis do. Thus, this principal is validated.

Principal 2

The proposed solutions perform both signing and verifying, and therefore the solutions satisfy the second principal.

Principal 3

Buypass, Commfides and MinID are intended to be used in the proposed solutions. Difi according to [19] has contract with the two former eID providers and MinID is developed by Difi. Therefore the proposed solutions satisfy the 3rd principal.

Principal 4

Standardized format for signing must be available, the proposed solutions uses the PAdES-LTV format, where the signature is embedded in the document. In addition long term validation is also added to the document in order to validate the signature in retrospect. Using this format satisfies the 4th principal.

Principal 5

The solution must cover the basic needs for signing, it means that the solution should make it possible to sign a document, but it must be minimum business logic in the signing process. With the proposed solutions it is possible to sign documents and there will not be any business logic in the signing process.

Principal 6

Service provider remains responsible for consideration; it means that the document must remain in service providers' system. I personally am not agree with this principal, because it will not satisfy a mutual signature model. Difi has thought an unilateral signature solution, where service provider has control over every thing. It is correct that service provider must be responsible for consideration, but remaining the document in service provider's system is not logical.

Principal 7

This principal states that the signature solution must follow the common principals for user interface. This means that the signature solution must support utilization of all possible eID providers existing in the country [19]. The proposed solutions support all three eID providers, where the user chooses which one to use, thus it satisfies this principal.

Principal 8

The verification must be adapted for self declared eID providers. It means that the condition that, eID provider must have contract with Difi in order to accept the signature, carried out by eID card issued by this eID provider, doesn't necessarily mean, not to accept signature carried out by a user using eID provider who do not have contract with Difi[19]. The proposed solutions do satisfy this principal, because the verification of a signature, carried out using PKI, will be supported. It is required that all necessary parameters for verification must be existed.

Principal 9

Verification service must be as a common component. The verification has three major operations, checking that the signature is cryptographically correct, checking that the eID used is issued by a valid eID provider and checking that eID and signature satisfy all other cryptographically requirements. The proposed solutions are validating the signature by using public key cryptography means. For validation of eID, all certificates involved in the signing process, are checked using two methods thus OCSP and CRL. Other cryptography requirements and quality of format is assumed to be of good quality. Therefore the proposed solutions satisfy this principal.

Principal 10

International verification of signature in long term is not considered of this thesis, but according to PEPPOL project all requirements set by PEPPOL are satisified by proposed solution of this thesis. PEPPOL project focuses on eID, based on PKI, and uses PAdES-LTV format, which is an ETSI standard. The proposed solutions of this thesis must support international verification of a signature in long term, because the same technologies are utilized in these solutions also.

6.2 Discussion

In this thesis a broad study of existing signature application, the solution proposal for common signature in the public sector of Difi, the technologies and standard to be used in Difi's pilot project, are carried out. Both existing solution and Difi's

proposed solutions are analyzed.

The analysis of Difi's proposed signature solution is presents in section 3.2.

Difi in the one hand claims that there is need for development of Advanced Electronic Signature, and on the other hand their proposed solutions do not satisfy the definition of advanced electronic signature.

The technologies utilized in the existing solution are described. The technology analysis indicates that there are weaknesses related in the technologies Difi intended to utilize.

Starting with PKI that PKI is intended to be utilized, but they are also weaknesses in PKI in which Difi do not considered. Weaknesses found in section 3.4 are reconsidered in order to cover these weaknesses in the proposed solutions of this thesis.

As mentioned previously that signing will be carried out using eID, namely Buypass or Commfides. Buypass today offers PKI based electronic ID, which is microprocessor based smart card with MULTOS as an operating system and RSA as an encryption algorithm. Private and public key, of length 1024 bit, pair is stored in this card. As mentioned the key with a length of 1024 bit is vulnerable within few years, but in the proposed solution of this thesis a key length of either 1536 or 2048 bit is considered in order to be secure in long term. According to [52] a key length of 2048 bit will give a lifelong security, i.e. it is not possible to be broken in our life. A 1536 bit key is consists of 436 decimal digits where a 2048 bit key is consists of 617 decimal digits. A 1536-bit key is fine within long time, but a 2048-bit key dominating in for most purposes. This is the main reason to choose 2048 bits key in order to be sure that a debenture letter is secured lifelong.

A disadvantage of using 2048 bits key is that the signing and verifying operations become slow.

In order to verify the signature a public key connected to private (the document is signed with) will be certified by the CA. In order trust signature the validation key must be certified. In the proposed solutions of this thesis, the document is not signed by eID card, but a new private key (called proxy private key) is derived from private key in eID smart card. The public key is connected to the proxy private key, must also be certified. The latter public key is certified by user which is called proxy certificate. Signing with proxy private key could be validated by both users public key (in smart card) and proxy signer's public key. This makes the trust model stronger such that the proxy certificate is certified by user, and user's certificate is certified by a well know CA, i.e. the user become a CA of his proxy signer, and thus will stand in certification authority path. This supports non-repudiation by the user (signer); whether he denies that he had not signed the document. As described in section 3.4 that hierarchical trust model for accepting

a public key as a valid key is preferred, because other ways of trusting is impossible.

For authentication of Altinn and Difi (in ID-porten) SSL/TLS certificates are used. The security of the SSL/TLS in existing solution (altinn) is not strong enough, because they use vulnerable algorithm (RC4) with a short length key of 128 bits and hash function (MD5) which is already broken. On the other hand Difi in ID-portal uses stronger security, and therefore the utility of the new proposal of Difi is better.

Authentication and encryption mechanisms intended to be used for proxy signer is to use SSL certificate, with AES-256 bit as a encryption algorithm, and SHA1 for computing a digest and RSA - 2048 bit as key exchange algorithm. The security of all three algorithms with the corresponding keys is strong enough. This will overcome the security shortcoming of existing signature solution.

As mentioned in section 3.4 that Difi is intended to use an applet based solution when signing by smart cards. This will create the" what you see what you sign" problematic, because the signer is seeing the document, but he cannot see whether he is signing the right document. This is overcome by this thesis's solution proposal that user (signer) is sending both the document and a signing order to his proxy signer in order to sign for him. So the user see what he is going to sign and therefore the "what you see what you sign" problematic is no longer existing. On the other hand the definition of advanced electronic signature will also be satisfied.

The storing format for signature is chosen to be PAdES-LTV by both the proposed solutions of Difi and this thesis. The main reason of choosing it by this thesis is that long term validation is in hand of signer. In other words user or service provider (chooses) how long validation of signature is required. The method of how to do it technically is described in section 2.5.3. The proposed solutions of this thesis satisfied all principals, except principal 6, set by Difi. As argued throughout this thesis, the proposed solutions covered all security shortcomings, found by this thesis. This will make the argument of that this thesis's proposed solutions are an improvement to existing solution and Difi's proposed solutions.

These arguments will support the claim that the proposed solutions of this thesis is an improvement to existing solution and Difi's solution proposals.

Chapter 7

Conclusion and Future Work

This chapter presents the achievements of this thesis. The project work was carried out as intended, but testing the claimed methods are not carried out because of time shortage. In addition remaining works to be done is also presented in this chapter.

7.1 Conclusion

In this thesis a broad study of existing signature application is done. The findings from this study showed that the existing signature application is vulnerable to some attacks. The signature application does not support long term validation. This solution is based on login, i.e. the login to the altinn and signs the document (sent as a form) by clicking (sign). It is also explored that this application is not user-friendly, i.e. it has low level of usability.

The proposed solutions of Difi have improved some of the security shortcoming of existing signature application. They have reused much of the existing signature application, and therefore many security flaws continued in the proposed solutions. Difi's proposed solutions intended to use asynchronous web services in one of the solutions and they argued that it will be easy to implement. On the other hand Difi assumed to use synchronous web services, where they again argued that it is difficult for service provider to implement it. It is because of time out that a web services' session will be killed after a short period of time. This make the proposal weak, because if it is difficult (or even not possible) to implement why it is assumed be utilized.

This thesis proposed three solutions in order to cover most needs. In the first and second solutions a service user and service provider model is utilized. In both solutions strong security parameters are assumed to be used. In the 1st solution the usability could be poor because putting heavy cryptographic computation (signature verification is done by the user) on users will make the system slow. On the

other hand the 2nd proposed solution carries out almost all heavy computation on proxy signer, where it is assumed that proxy signer is capable of these computations. But an evaluation of the first two solutions showed that the second solution, i.e. proxy centric signature solutions was preferred due to more advantages as described previously. These two solutions will cover the service providers' needs, but not when two private people want to sign a document. A 3rd solution proposal is designed to overcome this shortcoming, but to achieve better security an extra entity (Mail Archive) is added. Mail archive will make it possible to users' proxy signers to exchange the document in a secure way. The third solution is a real need in today's marked, because there is no such system designed or implemented yet. The security of this solution would be as strong as the two previously solutions.

As a main conclusion of achievements of this thesis; the proposed solution will improve both existing solution and Difi's proposed signature solution.

7.2 Future Work

As mentioned earlier that because of time shortage the proposed solutions are not tested. As a future plan the proposed solution must be tested and tried to be improved usability also. The main reason is that the end users'focal point is user friendliness and security. PGP failed in its usability test as claimed in [43]. The main reason was that people could not understand the system. In order to make the proposed solutions more user-friendly they must be developed to be as simple as possible.

Scalability is not considered in detail in this thesis which would be left to the future work. Scalability is also an issue in existing solution, because altinn was suffering of scalability this year when tax papers were sent out to the people. The server was overloaded several times due to heavy load, i.e. many users at the same time.

Bibliography

[1] A cryptovision whitepaper Version 1.0 (August 2009)cv cryptovision GmbH, "Side Channel Attacks on Smart Cards". Retrieved on 06.05.11, available at url: http://www.cryptovision.com/fileadmin/media/documents/Whitepaper_Produkte-/01-Whitepaper-Technical-Side-Channel_EN.pdf.

[2] Alexandra Boldyreva College of Computing, Georgia Institute of Technology, Atlanta, USA, Adriana Palacio Computer Science Department, Bowdoin College, Brunswich, USA and Bogdan Warinschi Computer Science Department, University of Bristol, Bristol, UK. "Secure Proxy Signature Schemes for Delegation of Signing Rights". Retrieved on 12.05.11, available at url: http://www.springerlink.com/content/g18rr14w0653w6n0/fulltext.pdf.

[3] Altinn II, "Funksjonell spesifikasjon – Sluttbrukerløsningen (SBL)". Retrieved on 10.02.11 from url: https://www.altinn.no/upload/6477/Funksjonell%20spesifikasjon%20-%20Sluttbrukerl%C3%B8sningen.pdf.

[4] Altinn, "Electronic channel for reporting to economic life, Implementation Guide, Enterprise system interface". Retrieved on 11.02.11, available at url: https://www.altinn.no/upload/1500/Fagsystem_Implementation%20Guide%2-0-%20English.pdf.

[5] Andreas Mauthe, Peter Thomas "Professional Content management system 2003" Addison and Wesley.

[6] Axel Martens, Humboldt-Universitat zu Berlin, Department of Computer Science, 10099 Berlin, Germany, "Usability of Web Services". Retrieved on 15.03.11 from url: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=1286801tag=1.

[7] B. Holcombe, Government smart card handbook., available at: http://www.smartcard.gov/information/smartcardhandbook.pdf.

[8] Brands Stefan A., "Rethinking Public Key Infrastructure and Digital Certificate: Building in Privacy" ISBN 0-262-02491-8.

[9] Bruce , Risk of Relying on cryptography: Inside Risks 112, Communications of the ACM, vol 42, n 10, Oct 1999. Retrieved on 31.05.11, available at url: http://www.schneier.com/essay-021.html.

[10] Buypass: Secuting Transaction "Key length: It is the size that applies". Accessed on 21.03.11, available at url: http://www.buypass.no/Bedrift/Hjem/5658.cms

[11] Buypass secuting transaction, "Smart Card technology". Accessed on 05.05.11, available at url: http://www.buypass.com/Home/Products+%26+services/Buypass+Smart+-Card/Technology.

[12] Buzzle.com, "Types of Smart Cards". Retrieved on 17.04.11 available at url: http://www.buzzle.com/articles/types-of-smart-cards.html.

[13] CNWIS-G2 Project eID "System Design Document" Version: 1.0, Revised on 04/09/2008 at Department of Computer Science and Engineering University of Moratuwa Sri Lanka. Retrieved on 13.04.11, available at ulr: http://codex.project-eid.org/site/images/c/cd/CS4200-08-DD-CNWIS-G2.pdf.

[14] CardWerk smart card solutions, "An overview of Secure Multi-Application Smart Card Operating System". Accessed on 05.05.11, available at url: http://www.cardwerk.com/smartcards/MULTOS/MULTOS_operating_syste-m_ovrview.aspx.

[15] Carliste Adnans And Steve Lloyd. Udrestanding PKI, Concepts, standards, and deployment considerations. Number ISBN 22(6)-672-32391-5 Adison-Wesley second edition, 2003.

[16] Chandana Gamage, Jussipekka Leiwo, and Yuliang Zheng, Peninsula School of Computing and Information Technology Monash University Australia "An Efficient Scheme for Secure Message Transmission using ProxySigncryptionn". Retrieved on 19.03.11.

[17] Clay Shirky O'REILLY "planning for web services obstacle and opportunities" april 2002 1st edition ISBN 0-596-0036.

[18] David Silverman. Doing quantitative research, chapter 10, page 140. SAGE publications, 2005.

[19] Difi, Directorate for ICT Management, "Overordnet arkitektur og prinsipper for utvikling av felles signeringsløsning" V0.9.6 –29.10.10 (Not published). Received from Tord I. Reistad Senior Advisor at Difi.

[20] Difi, Directorate for ICT Management, "Fellesløsning for signering i offentlig sektor" V1.0-11.02.2011(Not published). Received from Tord I. Reistad Senior Advisor at Difi.

[21] Digipost, Developed and operated by Posten Norge. "A digital mail Archive". Accessed on 18.05.11, available at url: https://www.digipost.no/privat/sp%C3%B8rsm%C3%A5l-og-svar.

[22] Do van Thanh, Dr. Scient.Professor - Senior Research Scientist, "Web Based Services and Semantics, Litcute notes spring 2010". Retrieved on 14.02.11, available at url: http://www.item.ntnu.no/academics/courses/ttm4128/lecture.

[23] ELMER 2 Standard,Forvaltningsinfo AS, 22. september 2005 for Nærings- og handelsdepartementet "Forslag til retningslinjer for brukergrensesnitt i offentlige skjemaer på Internet". Retrieved on 11.03.11 and available at url:http://www.elmer.no/retningslinjer/pdf/elmer2-endeligforslag.pdf.

[24] ETETSI Technical Specification 102 176-1 v.2.0.o (2007-11), "Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures;Part 1: Hash functions and asymmetric algorithms".

[25] ETSI TS 102 778-4 V1.1.1 July 2009, "Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term - PAdES-LTV Profile". Retrieved on 15.03.11 from url: http://www.etsi.org/deliver/etsi_ts/102700_102799/10277804/01.01.01_60/ts_10277804v010101p.pdf.

[26] ETSI Technical Specification 101 733 v.1.5.1 (2003-12), "Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats".

[27] Eric J. Braude, "Software Engineering: An Object-Oriented Perspectiv". John Wiley and sons INC. ISBN: 0-471-32208-3.

[28] Eric Newcomer , undrestanding web services; XML, WSDL, SOAP and UDDI isbn 0-201-75081-3 Addison and Wesley.

[29] Francisco Curbera, Matthew Duftler, Rania Khalaf,William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana IBM T.J.Watson Research Center, "Unraveling the Web Services Web. An Introduction to SOAP, WSDL, and UDDI".

[30] Fremantle, P., Weerawarana, S. and Khalaf, R. Enterprise services. Commun. ACM, 45 (10). 77-8.

[31] Gay Hardy, Zergo Ltd "The Truth Behind Single Sign-on Information Security Technical Report, Vol. 1, No. 2 (1996) 46-55" Retrieved on 08.03.11, available at url: http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6VJC-3VTK27G-7-1&_cdi=6091&_user=586462&_pii=S1363412797893569&_origin=gateway&_coverDate=12%2F31%2F1996_sk=999989997&view=c&wchp=dGLbVlz-zSkWA&md5=82667d9c528f6e2c930089b895d3df9c&ie=/sdarticle.pdf.

[32] Hamann, E.-M., Henn, H., Schack, T., Seliger, F., IBM Pervasive Computing Division, Boeblingen, Germany, "Securing e-business applications using smart cards". Rtrieved on 06.05.11, available at url: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5386931&abstractAccess=no&userType=inst.

[33] Hamed Taherdoos et. al. from Centre for Advanced Software Engineering and Faculty of Computer Science and Information Systems University Teknologi Malaysia, "Study of Smart Card Technology and Probe User Awareness about It: A Case Study of Middle Eastern Students". Retrieved on 12.03.11 available at url: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5234410.

[34] IBM Research, Center for Software Engineering, "Use Case Based Testing". Retrieved on 01.06.11, available at url: http://www.research.ibm.com/softeng/TESTING/ucbt.htm.

[35] Investigating Single Sign-on., Novell white paper, available at: http://www.novell.com/rc/docrepository/public/37/basedocument.2007-08-07.2321076507/4622014_en.pdf.

[36] International Standardization Organization ISO 32000-1:2008 first edition 01.07.08, "Document management — Portable document format — Part 1: PDF 1.7". Retrieved on 21.03.11, available at url: http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/PDF32000_2008.pdf.

[37] J. Aussel, Cards and Digital Identity., Telektronikk, Telenor, 2007, Volume 103, pp. 66-79.

[38] James Bean, 2009, SOA and Web Services Interface Design, Pricipals , Techniques and Standards", Printed in the US with ISBN:978-12-374891-1, 2009.

[39] Jianhong Zhang,Qianhong Wu, Jilin Wang, Yumin Wang, "An Improved Nominative Proxy Signature Scheme for Mobile Communication", Retrieved on 24.03.11, available at url:http://portal.acm.org/citation.cfm?id=977608.

[40] Jianhong Zhang,Qianhong Wu, Jilin Wang, Yumin Wang, An Improved "Nominative Proxy Signature Scheme for Mobile Communication", Key Laboratory of the Ministry Education, Xidian University , Xi'an 710071, P.R.China. Retrieved on 20.02.11, available at IEEE agazine at url: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=1283748.

[41] K. Mayes, K. Markantonakis, Smart cards, Tokens, Security and Applications., Springer Science+Business Media, 2008.

[42] K. Shum and Victor K.We, "A Strong Proxy Signature Scheme with Proxy Signer Privacy Protection". Department of Information Engineering, Chinese University of Hong Kong, Hong Kong. Retrieved on 20.02.11, available at IEEE magazine at url: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01029988.

[43] KA-PING YEE University of California, Berkeley, "Aligning Security and Usability". Received authumn 2010 (part of syllabus of specialization course TDT60 at IDI), is available at url: http://folk.ntnu.no/oztarman/tdt60/Aligning%20security%20and%20usability-.pdf.

[44] Lijang Yi, Gunqiaiig h i atid Guozhen Xian, "Proxy multi-signature scheme: A new typ of proxy signature scheme". Retrieved on 13.03.11, available at url: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=840145&tag=1.

[45] Meriram Webster "Word definations". Retrieved on 06.06.11, available at url: http://www.merriam-webster.com/dictionary/merchant.

[46] M. Lynne, Markus Daniel Robey. Producing consumable research about information systems. Information Resources Management Journal, 11:21, 1998.

[47] M. Myers VeriSign, R. Ankney CetCo, A. Malpani ValiCert, S. Galperin My CFO, C Adams Entrust Technologies June 1999. "Request for Comments: 2560, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP". Retrieved on 23.04.11 available at url: http://www.ietf.org/rfc/rfc2560.txt.

[48] MF1ICS70 Functional specification., Retrieved on 31.03.11, available at: http://www.nxp.com/acrobat_download/other/identification/M043541_MF1ICS70_Fspec_rev4_1.pdf.

[49] Masahiro MAMBO, Kelsuke USUDA and Eiji OKAMOTO, School of Information science, Japan Advanced Institute of Science and Technology "Proxy Signatures for Delegating Signing Operation" Retrieved on30.02.11, available at url: http://delivery.acm.org/10.1145/240000/238185/p48-mambo.pdf?ip=129.241.164.97&CFID=27907837&CFTOKEN=75463982__acm__=1307638585_04d05c2bfccdd1aa58ee77b92385e1b9.

[50] M. Shimaoka, Ed SECOM, N. Hastings NIST, and R. Nielsen Booz Allen Hamilton July 2008. Request for Comments: 3280 IETF "Memorandum for Multi-Domain Public Key Infrastructure Interoperability". Retrieved on 12.02.11, availabel at url: http://www.ietf.org/rfc/rfc5217.txt.

[51] MXC Software "Certificate Trust Model". Retrieved on 20.03.11, available at url: http://www.mxcsoft.com/Cryp_Trust%20Model.htm.

[52] NIST.org, "Answering Kaspersky Lab findings about weaknesses of RSA key length". Retrieved on 23.05.11 from url: http://www.nist.org/news.php?extend.259.

[53] Nick Pope, Thales Information Systems Security Aylesbury, United Kingdom, "Protecting Long Term Validity of PDF documents with PAdES-LTV". Retrieved on 26.04.11, available at url: http://www.springerlink.com/content/lm61112xv3724124/fulltext.pdf.

[54] Norwegian Law, "Law on Electronic Signature (eSignature)". Retrieved on 15.03.11, available at http://www.lovdata.no/all/tl-20010615-081-001.html1.

[55] OASIS, "Security Assertion Markup Language (SAML) V2.0 Technical Overview Working Draft 10, 9 October 2006". Retrieved on 23.03.11, available at url: http://www.oasis-open.org/committees/download.php/20645/sstc-saml-tech-overview-2%200-draft-10.pdf.

[56] OASIS, "Security Assertion Markup Language (SAML) V2.0 Technical Overview Working Draft 10, 9 October 2006". Retrieved on 23.03.11, available at url: http://www.oasis-open.org/committees/download.php/20645/sstc-saml-tech-overview-2%200-draft-10.pdf.

[57] OASIS Standard september 2003, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1. Retrived, available at url: (http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf).

[58] OASIS Working Draft 03, 20 February 2005, "Security Assertion Markup Language (SAML) 2.0 Technical Overview", Retrieved on 22.02.11, available at url: http://www.oasis-open.org/committees/download.php/11511/sstc-saml-tech-overview-2.0-draft-03.pdf.

[59] OASIS Standard 25 march 2008, "Security Assertion Markup Language (SAML) V2.0 Technical Overview Committee Draft 02. Retrived on 11.03.11, available at url: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf.

[60] OASIS Standards, "SSTC Response to "Security Analysis of the SAML Single Sign-on Browser/Artifact Profile" Committee Draft 01, 15 July 2005. Retrieved on 13.05.11, available at url: http://www.oasis-open.org/committees/download.php/13639/sstc-gross-sec-analysis-response-cd-01.pdf.

[61] Peter Henderson and Jingtao Yang, University of Southampton, Southampton SO17 1BJ, UK. "Reusable Web Services". Retrieved on 11.03.11, available at url: http://www.springerlink.com/content/kgx1g4mu83n0r234/.

[62] Professor Chris Reed "Journal of Information Law & Technology, JILT 2000 (3) - Chris Reed". Retrieved on 30.04.11 form url: http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2000_3/reed/.

[63] R. Housley RSA Laboratories, W. Polk NIST, W. Ford VeriSign, D. Solo Citigroup April 2002"Request for Comments: 3280 IETF: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile". Retrieved on 25.04.11, available at url: http://www.ietf.org/rfc/rfc3280.txt.

[64] Richard Ford and Michael Howard, "man-in-the-middle attack to the httPs Protocol". Retrieved on 02.03.11, available at url: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4768661&tag=1.

[65] Richard Mackey, at SearchSecurity.com "XML-based attacks and how to guard against them". Retrieved on 13.04.11, available at url: http://searchsecurity.techtarget.com/tip/XML-based-attacks-and-how-to-guard-against-them.

[66] Rongxing Lu, Zhenfu Cao, Xiaolei Dong Department of Computer Science and Engineering Shanghai Jiao Tong Universit and Renwang Su College of Statistics and Computing Science Zhejiang Gongshang University, "Designated Verifier Proxy Signature Scheme from Bilinear Pairings". Retrieved on 29.03.11, available at IEEE magazine url: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04673675&tag=1.

[67] S. Tuecke ANL, V. Welch NCSA, D Engret ANL, June 2004, "Request for Comments: 3820, Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile". Retrieved on 21.05.11, available at url:http://www.ietf.org/rfc/rfc3820.txt.

[68] SEID-Project "SEID-SDO-Data Object for Long term and Exchanging of electronic Signature", Version 1.0, 01.06.05. Retrieved on 11.05.11, available at url: http://www.npt.no/ikbViewer/Content/44963/SEID_Leveranse_3_v1.0.pdf.

[69] Scott Fluhrer Cisco Systems Inc, Itsik Mantin and Adi Shamir Computer Science department, The Weizmann Institute, "Weaknesses in the Key Scheduling Algorithm of RC4". Retrieved on 15.05.11, available at url: http://www.springerlink.com/content/w7fb0v5q582hxyrl/fulltext.pdf.

[70] Scott Isensee BMC Software nov. 4. 2002. "Usability of Web Services". Retrieved on 26.04.11, available at url: http://zing.ncsl.nist.gov/uig_w3c/IsenseeWebServicesUsability.ppt.

[71] SEI Project, Recommonded Certificate profiles for personal certificate and business certificate" Versiotn 1.02, 03.02.05. Retrieved on 11.05.11, available at url: http://www.npt.no/ikbViewer/Content/44963/SEID_Leveranse_3_v1.0.pdf.

[72] TechNet, Certfication Authority Trust Model". Retrived on 19.05.11, available at url: http://technet.microsoft.com/en-us/library/cc962065.aspx.

[73] Thomas Groß, IBM Zurich Research Laboratory, "Security Analysis of the SAML Single Sign-on Browser/Artifact Pro?le". Retrieved on 14.04.11, available at url: http://www.acsac.org/2003/papers/73.pdf.

[74] Thorsten Kleinjung et. al. "Factorization of a 768-bit RSA modulus". Received from Tord I. Reistad and can be retrieved, available at url: http://eprint.iacr.org/2010/006.pdf.

[75] VeriSign, a global Authentication Service Provider, "Secure Sockets Layer (SSL): How It Works". Accessed on 23.04.11 available at url: http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/index.html.

[76] William Stallings, Cryptography and Network Security: Principals and Practices, chapter 9, Fourth Edition, ISBN: 0-13-202322-9.

[77] William Stallings, Cryptography and Network Security: Principals and Practices, chapter 17.2, Fourth Edition, ISBN: 0-13-202322-9.

[78] Wade Trappe and Lawrences Washington, Introduction to cryptography with Coding Theory second edition, Peason Internation Edition ISBN 0-13-198199-4.

[79] W. Rankl, W. Effing, Card Handbook., 3rd Edition, Wiley, November 2003.

[80] What Makes a Smart Card Secure?., Smart Card Alliance Contactless and Mobile Payments Council White Paper, October 2008, available at: http://www.smartcardalliance.org/pages/download.

[81] w3school.com, "SOAP Introduction", Accessed on 18.04.11, available at url: http://www.w3schools.com/soap/soap_intro.asp.

[82] w3school.com, "SOAP HTTP Binding Toturial", Retrieved on 30.02.11, available at url: http://www.w3schools.com/soap/soap_httpbinding.asp.

[83] Wuwei Shen, Department of Computer Science, Western Michigan University, USA and Shaoying Liu Department of Computer Science, Hosei University, Tokyo, Japan "Formalization, Testing and Execution of a Use Case Diagram". Retrieved on 30.05.11, available at url: http://www.springerlink.com/content/vdwp45myh693rfbj/fulltext.pdf.

[84] Xiaoyun Wang and Hongbo Yu, Shandong University, Jinan 250100, ChinaHow to Break MD5 and Other Hash Functions , Eurocrypt 2005. Retrieved on 11.05.11, available at url: http://www.infosec.sdu.edu.cn/uploadfile/papers/How%20to%20Break%20MD5%20and%20Other%20Hash%20Functions.pdf.

[85] Y. Daniel Liang. Introduction to java Programming, sixth eddition Pearson 2006, ISBN: 0-13-222158-6.

Appendices

The appendics present the abstract theories of related technologies, not mention in over, in this thesis.

INDEX A

This appendex present SAML components, taken from [56]. The SAML components and their individual parts are as follows:

- Assertions: SAML allows for one party to assert characteristics and attributes of an entity. For instance, a SAML assertion could state that the user is "John Doe", the user has "Gold" status, the user's email address is john.doe@exampl-

  e.com, and the user is a member of the "engineering" group. SAML assertions are encoded in a XML schema. SAML defines three kinds of statements that can be carried within an assertion:
- Authentication statements: are issued by the party that successfully authenticated the user. They define who issued the assertion, the authenticated subject, validity period, plus other authentication related information.
- Attribute statements: contain specific details about the user (for example, that they have "Gold" status).
- Authorization decision statements: identifies what the user is entitled to do (for example, whether he is permitted to buy a specified item).
- Protocols: SAML defines a number of request/response protocols. The protocol is encoded in an XML schema as a set of request-response pairs. The protocols defined are. Sstc-saml-tech-overview-2.0-draft-03 20 February 2005 Copyright © OASIS Open 2004. All Rights Reserved. Page 7 of 40
- Assertion Query and Request Protocol: Defines a set of queries by which existing SAML assertions may be obtained. The query can be on the basis of a reference, subject or the statement type.
- Authentication Request Protocol: Defines a ¡AuthnRequest¿ message that causes a ¡Response¿ to be returned containing one of more assertions pertaining to a Principal. Typically the ¡AuthnRequest¿ is issued by a Service Provider with the Identity Provider returning the ¡Response¿ message. Used to support the Web Browser SSO Profile.
- Artifact Protocol: Provides a mechanism to obtain a previously created assertion by providing a reference. In SAML terms the reference is called an "artifact". Thus a SAML protocol can refer to an assertion by an artifact, and then when a Service Provider obtains the artifact it can use the artifact Protocol to obtain the actual assertion using this protocol.
- Name Identifier Management Protocol: Provides mechanisms to change the value or format of the name of a Principal. The issuer of the request can be either the Service Provider or the Identity Provider. The protocol also provides a mechanism to terminate an association of a name between an Identity Provider and Service Provider.

- Single Logout Protocol: Defines a request that allows near-simultaneous logout of all sessions associated by a Principal. The logout can be directly initiated by the Principal or due to a session timeout.
- Name Identifier Mapping Protocol: Provides a mechanism to enable "account linking". Refer to the subsequent sections on Federation.
- Bindings: This details exactly how the SAML protocol maps onto the transport protocols. For instance, the SAML specification provides a binding of how SAML request/responses are carried with SOAP exchange messages. The bindings defined are:
- SAML SOAP Binding: Defines how SAML protocol messages are transported within SOAP 1.1 messages. In addition it also defines how the SOAP messages are transported over HTTP.
- Reverse SOAP (PAOS) Binding: Defines a multi-stage SOAP/HTTP message exchange that permits a HTTP client to be a SOAP responder. Used in the Enhanced Client and Proxy Profile and particularly designed to support WAP gateways.
- HTTP Redirect Binding: Defines how SAML protocol messages can be transported using HTTP redirect messages (i.e. 302 status code responses)
- HTTP POST Binding: Defines how SAML protocol messages can be transported within the base64-encoded content of an HTML form control
- HTTP Artifact Binding: Defines how a reference to a SAML request or response (i.e. an artifact) is transported by HTTP. Defines two mechanisms, either an HTML form control, or a query string in the URL.
- SAML URI Binding: NOT SURE HOW TO EASILY DEFINE THIS
- Profiles: The core of the SAML specification defines how the SAML requests and responses are transported, however, a number of use cases have been developed that require the formulation of Profiles that define how the SAML assertions, protocols and bindings are combined. Some of these described in detail later on in the document, in summary they are:
- Web Browser SSO Profile: Defines how a Web Browser supports SSO, when using ¡AuthnRequest¿ protocol messages in combination with HTTP Redirect, HTTP POST and HTTP Artifact bindings.
- Enhanced Client and Proxy (ECP) Profile: Defines how ¡AuthnRequest¿ protocol messages are used when combined with the Reverse-SOAP binding (PAOS). Designed to support mobile devices front-ended by a WAP gateway.
- Identity Provider Discovery Profile: Defines how a service provider can discover which identity providers a principal are using with the Web Server sstc-saml-tech-overview-2.0-draft-03 20 February 2005 Copyright © OASIS Open 2004. All Rights Reserved. Page 8 of 40.
- Single Logout Profile: A profile of the SAML Single Logout protocol is defined. Defines how SOAP, HTTP Redirect, HTTP POST and HTTP Artifact bindings may be used.

- Name Identifier Management Profile: Defines how the Name Identifier Management protocol may be used with SOAP, HTTP Redirect, HTTP POST and HTTP Artifact bindings.
- Artifact Resolution Profile: Defines how the Artifact Resolution protocol uses a synchronous binding, for example the SOAP binding.
- Assertion Query/Request Profile: Defines how the SAML query protocols (used for obtaining SAML assertions) use a synchronous binding such as the SOAP binding.
- Name Identifier Mapping Profile: Defines how the Name Identifier Mapping protocol uses a synchronous binding such as the SOAP binding.