EDL(s)

Electronic Driving License(s)

To increase traffic safety and improve other functions vital to society by implementing and deploying an electronic driving license (EDL) framework.

Computing Science and Engineering
Umeå University

(Institutionen för datavetenskap
Umeå universitet)

2015-04-06

Mattias Hållström
# Table of Contents

## Chapter 1 Introduction

1.1 Background ................................................................. 9
1.2 Problem statement ......................................................... 10
1.3 Method ........................................................................ 10
1.4 Purpose ......................................................................... 10
1.5 Previous research ........................................................ 11
  1.5.1 Fred Goldberg and EDL system KitteLock .................. 11
1.6 Today’s usage ............................................................... 14

## Chapter 2 Road safety statistics

2.1 Need for electronic driving license .................................... 15
2.1 European and Swedish goals regarding road safety ........... 17

## Chapter 3 Laws and regulations

3.1 Driving licenses and unlicensed driving ............................ 19
3.2 Digital signatures .......................................................... 21
  3.2.1 Different types of electronic signatures ..................... 21
  3.2.2 Qualified electronic signatures ................................. 22
  3.2.3 Secure signature-creation devices ........................... 22
  3.2.4 Qualified certificates ............................................. 22
  3.2.5 Legal validity of electronic signatures .................... 23

## Chapter 4 Technology

4.1 Smart cards .................................................................. 25
  4.1.1 Data storage and data zones .................................. 27
  4.1.2 Form factor .......................................................... 31
4.2 Authentication ............................................................... 31
  4.2.1 Passwords ............................................................ 32
  4.2.2 Public-key cryptography ......................................... 32
  4.2.3 Biometrics ........................................................... 33
4.3 Cryptography and PKI ...................................................... 33
  4.3.1 Digital certificates .................................................. 35
  4.3.2 Certification authority (CA) .................................... 36
  4.3.3 Registration authority (RA) .................................... 36
  4.3.4 Certificate repository ............................................ 37
  4.3.5 Revocation of digital certificates ............................. 37
  4.3.6 Format of an X.509 digital certificate ..................... 37
  4.3.7 Digital signing ..................................................... 46
  4.3.8 PKI asymmetric data encryption ............................ 48
  4.3.9 Key lengths ........................................................ 49
4.4 Vehicle technology ......................................................... 49
  4.4.1 Controller area network (CAN) bus ......................... 49

## Chapter 5 Electronic driving license (EDL)

5.1 Merging information ..................................................... 51
5.2 Form factor and visible contents of European driving licenses 51
5.3 EDL information contents .............................................. 54
5.4 XML format – EEDL ...................................................... 56
  5.4.1 Document number ................................................ 57
  5.4.2 Issuing date .......................................................... 57
  5.4.3 Expiry date .......................................................... 57
  5.4.4 Issuing authority ..................................................... 57
| 5.4.5  | ICE – In case of emergency | 58 |
| 5.4.6  | Identification             | 58 |
| 5.4.7  | Address                    | 58 |
| 5.4.8  | Driving license             | 59 |
| 5.4.9  | Medical information         | 60 |
| 5.4.10 | Traffic insurance           | 61 |
| 5.5    | Storage and access control  | 61 |
| 5.5.1  | EDL key and certificate file storage and access | 62 |
| 5.5.2  | EDL information file storage and access | 64 |
| 5.6    | Digital certificate extensions | 65 |

**Chapter 6**  
Driver and vehicle verification service (DaV-VS) ............................................................... 66

6.1 Verification system .................................................................................. 66
6.2 Driver and vehicle database (DaV-DB) .................................................. 66

**Chapter 7**  
Electronic driver license (EDL) verification ........................................................................ 68

7.1 Driver authentication .................................................................................. 69
  7.1.1 Password ......................................................................................... 69
  7.1.2 Biometrics ...................................................................................... 69
  7.1.3 PKI ................................................................................................. 70
7.2 Vehicle authentication .................................................................................. 70
7.3 Online EDL verification .............................................................................. 71
  7.3.1 Permanent network connection ....................................................... 72
  7.3.2 Nonpermanent network connection ............................................... 77
7.4 Offline EDL verification ............................................................................. 79
  7.4.1 On-card offline license verification ................................................. 80
  7.4.2 On-board offline license verification ............................................. 82
7.5 Online/offline adaptable EDL verification ................................................ 83
  7.5.1 Online access and updated local database ..................................... 84
  7.5.2 Online access and outdated local database .................................... 85
  7.5.3 Offline access and updated local database .................................... 85
  7.5.4 Offline access and outdated local database ................................... 86
  7.5.5 Change in online/offline access .................................................... 87
7.6 Update of local database ............................................................................ 87
7.7 Negative validations ................................................................................... 87

**Chapter 8**  
EDL framework usage ...................................................................................... 89

8.1 Information advantages .............................................................................. 89
8.2 Reduce drunk driving ............................................................................... 89
8.3 Acknowledgement of misuse .................................................................... 90
8.4 Drive log ................................................................................................. 91
8.5 Advantages in police work ....................................................................... 91
  8.5.1 Electronic issuing of tickets ........................................................... 92
8.6 Driving with a learner’s permit ................................................................. 92
8.7 Driving with a lost valid driving license ................................................ 93
8.8 Driving in emergency situations .............................................................. 93

**Chapter 9**  
Privacy and information security ...................................................................... 94

9.1 PKI risks ................................................................................................. 94
9.2 Drive logs and other information ............................................................ 94
9.3 Manipulating vehicles ............................................................................. 95
9.4 Biometrics and privacy .......................................................................... 96
9.5 Offline license verification issues .......................................................... 96
9.6 Law enforcement information ............................................................... 96
9.7 Medical information on EDL ................................................................. 97
9.8 What you see is what you sign ................................................................ 97
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10.1</td>
<td>Report results and influence on future development</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>References</td>
<td>102</td>
</tr>
<tr>
<td>12</td>
<td>12.1</td>
<td>EDL XML (EF_{EDL})</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>12.2</td>
<td>EDL XML schema</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>12.1</td>
<td>Additional driving license information and restriction(s)</td>
<td>119</td>
</tr>
</tbody>
</table>
Abstract

Researcher Fred Goldberg closed a report by expressing, “It is now up to the politicians to create a safer traffic environment by utilizing new technology for the benefit of society” (Goldberg, 1995). And as he stated prior to that quote, a technique is available to prevent unlawful driving, to effectively reduce drunken driving and, among other things, to reduce theft of cars. This report presents the available technology and design that make it possible to incorporate an electronic driving license (EDL) framework into everyday life. This report will focus on how such technology could be used to prevent drivers from driving without valid driving licenses and to improve traffic safety in numerous other ways.

The report presents the existing laws and regulations that govern the area of usage and describes how such legal aspects affect the design and deployment of an EDL framework. The presented legislation is extracted primarily from the European Union in general, and as a national example, Swedish legislation is often used. Moreover, the report will present technology that could be used to implement an EDL framework using examples of implementations and design, including license verification and an EDL. The implementation and deployment of an EDL framework could result in privacy concerns, and such aspects are discussed in a chapter where it is stated that security and privacy issues must be considered, as the potential for misuse is great. The level of privacy must be analysed in relation to the lifesaving potential of such a framework.

Preface

A departure from my studies in computer science, my main profession is working as a police sergeant in Skellefteå, Sweden. During my service years, since 2008, I have numerous times been in contact with drivers without a valid driving license or whose driving license has been revoked. My experiences, I am sorry to say, include situations where human lives could have been saved if drivers without a valid driving license could have been stopped from driving.

The technology of today could provide us with the proper tools to save numerous lives around the world by preventing unlicensed drivers from even starting vehicles they have not been issued a license to drive. This report will present an electronic driving license (EDL) framework that highlights the technology behind, as well as the design of, EDLs and their required verification methods.

I am inspired by the previous research done by the Swedish researcher Fred Goldberg, and the framework presented within this report is in many ways a development of his initial research results.

The report will also include a chapter discussing the aspects of privacy and information security as the usage of EDLs has the potential to violate people’s privacy if information would be misused.

Acknowledgements

I would like to thank the Swedish researcher Fred Goldberg for his presentation and discussion regarding his previous research and his thoughts about the future of EDLs.

His initial research and dedication form a foundation suitable as a starting point for more research and provide readers with the image of how an EDL framework would improve traffic safety and should be incorporated in some form into today’s traffic environment.

I would also like to thank Jan Erik Moström, my supervisor at Umeå University, for his comments and helpful suggestions for the report.
Umeå, April 2015
Mattias Hållström
Chapter 1 Introduction

1.1 Background

Road safety work in Sweden is based primarily on the declaration that no one should ever have to be killed in a traffic accident. This goal is called Vision Zero (“Nollvisionen”) (Vägverket, 2008) and was approved by the Swedish Parliament in 1997. Road safety work is divided into interim objectives with the purpose of continuously, year by year, reducing the number of fatal accidents and also the number of people seriously injured in traffic accidents. A crucial part of road safety work to prevent traffic accidents is to stop people from driving vehicles for which they do not have the required driving license. Swedish road safety work is largely about handling human errors, which are bound to happen as long as there are humans manoeuvring vehicles. The occurrence of human errors is therefore one aspect that has to be managed to minimize their negative impact on road safety. If technology can prevent some human mistakes or misbehaviours, we are well on our way to a much safer society regarding road traffic.

Every year there are many traffic accidents involving drivers that do not hold the required driving license for the vehicle category they were driving at the time of the accident. Such accidents could, and in many cases do, result in both severe injuries and fatalities.

Today’s technology should, with the appropriate implementation, give us the tools to avoid many of these accidents by preventing unlicensed drivers from even starting a vehicle they are not allowed to drive. The introduction of an electronic driving license (EDL) scheme connected to the ignition of vehicles would guarantee that only drivers with valid licenses for that motor vehicle category would be able to start them.

An implementation of the EDL scheme could thus be used to prevent motor vehicle theft and other unauthorized usage of such vehicles.

The above figure does not show an encrypted version of the alphabet. These letter and digit combinations are driving license vehicle categories that were defined by the European Parliament through a directive (2006/126/EC) issued in 2006 (the European Union, 2006) and since then gradually integrated into the member states’ legal systems. The Swedish government included the final vehicle categories into the driving license law\(^1\) as of January 2013. A license including one or more of these defined vehicle categories gives the holder the right to drive any vehicle in those categories.

When a police officer or some other interested party does a validity check of a driving license, the physical license itself often is not a guarantee that it is valid. Driving licenses can be counterfeited or revoked; one person might be holding a license that belongs to another person. Thorough controls by means of database searches in various registers should, in most cases, reveal such attempts to drive

\(^1\) Swedish act on driving licenses – “Körkortslag” (SFS 1998:488).
vehicles without the proper driving license. But many times the only control of a driving license is the visual check done by a police officer, who checks that the photo resembles the person driving the vehicle and that the vehicle conforms with the license’s vehicle categories.

How would it be if people who lacked the license to drive a certain vehicle type wouldn’t be able to even start such vehicles? Today’s technology allows us to implement such a solution. Why aren’t these technical solutions used today as an everyday way of life, and maybe even more importantly, how would such technical solutions work?

1.2 Problem statement

Prototypes of EDLs have previously been developed and tried out in Sweden by the Swedish scientist Dr. Fred Goldberg. This report tries to answer questions like:

Why do we need to deploy an EDL framework?

Will this technology reach the main public and be a part of our everyday life?

What are the possibilities and difficulties regarding both technology and legal aspects? Can we combine the technology and legal aspects to make way for an everyday use?

How does today’s technology work, and what are the options?

How should an EDL framework be designed considering EDL design and verification?

How do EDLs affect personal privacy, and what do developers have to consider?

How will technology allow approved unlicensed driving, such as driving with a learner’s permit?

What are the possibilities regarding crime prevention work as a positive effect of future EDL technology in vehicles?

1.3 Method

The problem statement will be answered by literature studies and theoretical prototypes. The literature studies will, first and foremost, target areas of study such as data storage, data processing, data communication and data security to present theoretical prototypes and scientifically answer the questions raised above. Furthermore, these literature studies will focus on legal aspects that regulate the various possibilities for design and implementation of these theoretical prototypes of EDL systems connected to vehicle ignition.

Accordingly, this paper presents theoretical implementations, and therefore none of the presented prototypes or technologies are implemented in the real world.

1.4 Purpose

The purpose of this report is to analyse the difficulties and present the possibilities of deploying a framework for EDLs. The report will bring together previous research, legal aspects and existing technology and include future EDL design to present an EDL framework that would be deployable in today’s society and vehicle industry. The EDL framework will be focused mainly on European, and in many aspects more specifically Swedish, legislation and requirements regarding driving licenses and road traffic security. The implementation and deployment of a complete EDL framework could perhaps pose a threat to license holders’ privacy, so a chapter about privacy has been included.
1.5 Previous research

1.5.1 Fred Goldberg and EDL system KitteLock

During the late nineties, and until the early start of the following decade, a Swedish researcher by the name Dr. Fred Goldberg was developing a prototype of an EDL system that was field tested in Sweden. His research began after his stepdaughter was tragically killed by a drunken driver who was driving a car while holding a suspended driving license. The research was conducted together with the Swedish road administration and also with the two vehicle companies Volvo and Saab. The EDL system was named KitteLock, after his stepdaughter, and the EDL was based on a smart card with a built-in microchip. The smart card contained information regarding the driver and the driver restrictions. The EDL card was meant to replace the traditional ignition key. The system was designed as an EDL card that was inserted into an EDL computer installed in the vehicle (Goldberg, 1995).

Fred Goldberg, who was an innovator in this field of work, proposed that the EDL could contain the following information, in addition to the regular driving license information (Goldberg, 1999):

- Driving time for commercial drivers
- Traffic insurance information
- Encrypted fingerprint information
- Colour photograph of the driver
- Language code of the driver
- Memory area for driver- and/or vehicle-specific information
- Medical information
- Organ donor information
- Name and address of doctor who knows the driver’s medical history
- Name and address of relatives to be contacted if the driver is injured or killed

---

Vägverket.
Fred Goldberg concluded that an EDL system would help stop or reduce unlicensed driving, drunk driving, car theft, criminal activity, the use of uninsured cars, overtime driving and dangerous car chases. Moreover, the KitteLock system, after further development, would make it possible to integrate automatic alarm systems and facilitate the use of cars by handicapped persons.

Fred Goldberg’s previous research regarding EDLs used an offline driving license verification approach in which the driving license information was stored inside the smart card. Fred Goldberg’s initial EDL cards, such as those from 1993, were implemented via contact smart cards. He soon discovered the drawbacks of using contact smart cards: the wear and tear on the contact surfaces could lead to occasional misreadings that sometimes led to the whole system being jammed. As a result of these problems with contact smart cards, Goldberg changed his approach and began implementing EDLs with contactless smart cards (Goldberg, 1999).

Goldberg proposed that the on-board EDL driving monitor could be extended to serve as an electronic driving monitor. He compared the function of such an EDL driving monitor with that of the black box of an airplane (Goldberg, 1995).

Fred Goldberg also included, during the Swedish field tests in 1997, the ability to remotely stop vehicles using the KitteLock EDL system. The receiver was a paging unit that was integrated with the system electronics. The remote stopping process was initiated by sending a secret code to the paging unit. If the code matched a predetermined code within the system, the vehicle stopping process was initiated. In this process, a message was presented on a display to the driver, informing him or her that the vehicle would be stopped in 30 seconds. The countdown sequence was displayed continually until the countdown indicator reached 0 seconds and the engine was switched off. In addition, the hazard lights were turned on at the start of the countdown sequence to warn surrounding traffic. There was also an acoustic signal inside the car so the driver would not miss the warning (Goldberg, 1999).
As a positive effect of implementing EDLs, thus eliminating unlicensed driving, the impacts would include the following (Goldberg, 1998):

- The number of intoxicated (by alcohol or other drugs) drivers would be reduced, as many of these drivers are repeat offenders with a suspended driving license.
- The number of drunken driving fatalities would be reduced.
- Total CO₂ emissions would be reduced.
- The number of stolen vehicles would be strongly reduced if the EDL system were integrated with the electronic engine control through a CAN bus or similar. This would also have an effect on crimes such as bank robbery, burglary and smuggling as such crimes often are committed using stolen vehicles.

Fred Goldberg, in cooperation with Swedish National Railways (SJ), also installed a test system in an ASEA RC Locomotive 1996. The locomotive driving license stored a code for identifying which locomotive type the train driver was licensed to drive.

Fred Goldberg ended one of his presentations with the text “You have now seen a glimpse of the future”. He saw the potential of the deployment of an EDL system and worked towards acceptance from the general public as well as from politicians. Had politicians supported the future implementation and deployment of EDLs perhaps many human lives could have been saved.

---

3 “Du har nu fått en glimt av framtiden.”
1.6 Today’s usage

To some extent the partial usage of EDLs has begun spreading across the world. We can see examples such as the MyKad in Malaysia, the new Gemalto driving license in Mexico and the Belgium eID. This partial use of the full potential offered by a complete EDL system, such as the one presented by Dr. Fred Goldberg, does not prevent an unlicensed driver from driving vehicles. The examples in use today merely convert the driving license into an EDL, thus lacking many of the benefits of a complete EDL system that electronically checks the driver’s license to verify that he or she is permitted to start the vehicle. If a full EDL framework was deployed unlicensed drivers or drivers unqualified for the vehicle category would not be able to start the vehicle. A full EDL system could be implemented with both online and offline verification features, as will be shown later in this report, to ensure even better reliability.
Chapter 2 Road safety statistics

2.1 Need for electronic driving license

In 2011, about 78,000 traffic violations were reported in Sweden, and many of them, about 35,000, were cases of driving without a valid driver’s license (Justitiedeparmentet, 2012, p. 42). Many thousands of traffic violations that may result in accidents and fatal crashes occur every year across Sweden. These numbers do not include the drivers that never get reported and therefore avoid prosecution. The diagram below shows the development in Sweden since 1975 in total reported traffic violations and the number of people reported for driving without a valid driver’s license.

![Diagram showing reported crimes in Sweden from 1975 to 2013](image)

*Figure 2 - Reported crimes (unlicenced driving and all traffic crimes) in Sweden during the years 1975 to 2013*

The diagram indicates a small decrease in reported offences regarding only unlicensed driving in Sweden the last 10 years. But overall, over the long term since 1975, the number of reported offences shows no sign of major reduction.

An in-depth study performed by the Swedish Transport Administration (Trafikverket) between 2009 and 2011 revealed that 25% of motorcycle drivers killed in traffic accidents didn’t have a valid driving license in vehicle category A. And at a negative peak in 2007, almost 40% of deadly motorcycle accidents involved motorcyclists without a valid driver’s license in vehicle category A. In July of 2013,
Swedish television reported that during the first six months of that year, five out of six motorcycle drivers killed in accidents were driving without a valid driver’s license (Björk, 2013).

From an international perspective, from 2007 to 2009, nearly one in five (18.2%) of all fatal crashes in the United States involved an unlicensed or invalidly licensed driver (Foundation for Traffic Safety, 2011, p. 13). These fatal crashes resulted in 21,049 deaths.

Furthermore, this American report concludes that unlicensed drivers were almost 10 times (9.4) more likely to flee (leave the scene of the crash) compared to validly licensed drivers. Or in comparison, it is estimated that over 50% of all drivers leaving the scene of a crash were unlicensed or invalidly licensed drivers. The report also shows a relationship between drivers involved in fatal crashes and drunk driving: nearly half (49.8%) of the drivers involved in fatal crashes had some amount of alcohol in their system. Finally, about one third of illegal drivers were under the age of 20.

Similar results are presented by the Australian government, showing that between 10% and 20% of all fatal crashes involve unlicensed motorists (Austroads Ltd, 2013). They also point out that EDLs could be one way of reducing unlicensed or invalidly licensed driving.

A science report from the city of Auckland (New Zealand) observes that unlicensed drivers had a significantly higher risk of car crash injury than those holding a valid license. The risk was about 11 times greater for unlicensed drivers, after adjustments for age and sex (Stephanie Blows, 2005, pp. 230-234).

Unlicensed driving or invalidly licensed driving can be a result of many factors:

- Never obtaining a driving license
- Driving with revoked driver’s license (for example, after being convicted of speeding)
- Driving wrong vehicle category (for example, driving a motorcycle without a vehicle category A license)
- Not following license restrictions (for example, not wearing glasses or contact lenses despite having condition code 01.06 included on the driving license)
- Driving with nonvalid foreign driving license

All of the above could be addressed by EDLs together with appropriate vehicle electronics and control functions, such as EDLs connected to the vehicle ignition to prevent unlicensed or invalidly licensed driving. Other approaches could be control mechanisms that indicate whether the driver in a vehicle presented a valid driving license on start.

Earlier technological advances throughout the vehicle industry clearly show how technology can reduce criminal activity. A worthy example is how the European Union’s demand for a mandatory immobilizer in all new cars sold within the member states correlates to a major reduction in the number of vehicle thefts. The risk of being a victim of car theft in Sweden hasn’t been this low since the 1970s (see diagram below).
Figure 3 - Reported car thefts in Sweden during the years 1975 to 2013 and the introduction of mandatory immobilizers

The mandatory use of EDLs connected to vehicle ignition would most probably save numerous lives and prevent many people from being seriously injured. Based on the research, we can assume that at least 10% of all people killed in traffic road accidents could be saved if every driver has to have a valid driver’s license to start a vehicle. Mandatory use of EDLs making it impossible to start a vehicle without a valid driving license would accordingly have saved over 110,000 people between 1991 and 2011, when the total number of persons killed was over 1.1 million. In Sweden, where the technology was somewhat ready for deployment as of the mid-1990s, mandatory EDLs connected to vehicle ignition could have saved, as calculated above (1995–2011), over 800 lives and prevented over 35,000 people from getting slightly or severely injured (Trafikanalys, 2012).

In Sweden in 2005 the price tag on road traffic accidents was 20.9 billion SEK, or about 2.3 billion Euros (MSB, 2011). Should the same calculations be applied to these economic numbers, we can clearly see the potential for substantial resource savings. It is a significant saving of resources even in a small country like Sweden, where traffic safety is considered to be among the best in the world.

2.1 European and Swedish goals regarding road safety

The European Union’s road safety guidelines aim to cut European road deaths by 50% from 2010 to 2020 by making users, vehicles and infrastructure safer (European Commission, 2010). As a reference, 35,000 people died in traffic accidents in 2009, and over 1.7 million people were injured during the same year.

In 1997, the Swedish parliament decided to work towards the national goal called Vision Zero (Vägverket, 2008). This national goal was for no one to ever be killed in a traffic accident in Sweden. As
a partial goal, the Swedish Parliament decided in 2009 that between 2007 and 2020, Swedish road deaths should be reduced by 50% and the number of those seriously injured by 25% (Vägverket, 2008).
Chapter 3 Laws and regulations

3.1 Driving licenses and unlicensed driving

Most countries, including Sweden and every other European Union (EU) member country implementing directives issued by the European Parliament, require that every driver complete a driving test to be issued a driving license. Throughout the world the regulations regarding driving licenses and the layout and languages of the physical license differ greatly. Member countries of the EU are undergoing a harmonization regarding these areas. EU standardization (the European Union, 2006) has defined the following vehicle category codes for driving licenses in the member countries:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Power Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Moped</td>
<td>≤ 50 cc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 45 km/h</td>
</tr>
<tr>
<td>A1</td>
<td>Motorcycle (very limited effect)</td>
<td>≤ 125 cc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 11 kW</td>
</tr>
<tr>
<td>A2</td>
<td>Motorcycle (limited effect)</td>
<td>≤ 25 kW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 0.16 kW/kg</td>
</tr>
<tr>
<td>A</td>
<td>Motorcycle</td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>Tricycles or quadricycles (optional, not used in Sweden)</td>
<td>≤ 550 kg</td>
</tr>
<tr>
<td>B</td>
<td>Private car or light lorry (with light trailer)</td>
<td>≤ 3,500 kg</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 8 passengers</td>
</tr>
<tr>
<td>C1</td>
<td>Car or lorry (with light trailer)</td>
<td>3,500–7,500 kg</td>
</tr>
</tbody>
</table>

Table of vehicle category codes is a simplified version of the complete definition in Directive 2006/126/EC of the European Parliament and of the Council.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Car or lorry (with light trailer)</td>
<td>$\geq 3,500$ kg</td>
</tr>
<tr>
<td>D1</td>
<td>Bus (with light trailer)</td>
<td>$&gt; 8$ passengers, $\leq 16$ passengers</td>
</tr>
<tr>
<td>D</td>
<td>Bus (with light trailer)</td>
<td>$&gt; 8$ passengers</td>
</tr>
<tr>
<td>BE</td>
<td>Car or light lorry (tractor vehicle category B)</td>
<td>$&gt; 8$ passengers</td>
</tr>
<tr>
<td>C1E</td>
<td>Car or lorry (tractor vehicle category B)</td>
<td>$3,500$–$7,500$ kg, Heavy trailer</td>
</tr>
<tr>
<td>CE</td>
<td>Car or lorry</td>
<td>$\geq 3,500$ kg, Heavy trailer</td>
</tr>
<tr>
<td>D1E</td>
<td>Bus</td>
<td>$9$–$16$ passengers, Heavy trailer</td>
</tr>
<tr>
<td>DE</td>
<td>Bus</td>
<td>$&gt; 8$ passengers, Heavy trailer</td>
</tr>
</tbody>
</table>

The European Parliament states in its directive on driving licenses that driving licenses should be issued only to one who has passed a test of skills, test of behaviour and theoretical test and meets the medical standards in accordance with the directive. The directive aims to harmonize driving license regulations throughout the EU and, in many cases, clarifies only requirements that already exist within most member states’ legal systems.

The Swedish driving license law states, in its second chapter, first paragraph (English translation):

*Car, lorry, bus, motorcycle, moped class I, ATV, heavy equipment class I may only be driven by one who has a valid driving license for such a vehicle. (SFS (1998:488))*

Driver’s license laws differ around the world, as do the penalties for unlicensed driving. Unlicensed driving can result in fines and/or imprisonment. Some legal systems allow for vehicle confiscation in cases of unlicensed driving.
The Swedish Road Traffic Offences Act states, in its third paragraph (English translation):

*If someone, with intent, drives a vehicle requiring a driver’s license without having the correct license, he will be sentenced for unlicensed driving (unlawful driving). If he previously had a license that has been revoked or if the crime occurred routinely or is in any other way regarded as a felony, the sentence is imprisonment up to six months.*

### 3.2 Digital signatures

The Swedish law regarding qualified digital or electronic signatures, the Act on Qualified Electronic Signatures (SFS (2000:832)), was introduced 1 January 2001 and based on the Directive of the European Parliament and the Council on a Community framework for electronic signatures (the European Parliament, 1999). The directive gives rules regarding electronic signatures and certificate services with the goal of coordinating the technical and legal work throughout the EU’s member states. The directive’s main purpose is to clear obstacles regarding the EU’s inner market by primarily facilitating and otherwise increasing security concerning the electronic trade market and guaranteeing that electronic signatures could be used throughout the EU. This would therefore facilitate digital signing across member states’ borders (Cardholm, et al., 2001, pp. 19-20). The directive was passed 30 November 1999 and took effect 19 January 2000. It was to be implemented in the member states by 19 July 2001 at the latest. Thus, Sweden implemented the directive early (Sjöberg Magnusson, 2000-2001, pp. 864-882).

The Swedish law on qualified electronic signatures applies to electronic signatures, signature-creation devices and certificates.

The purpose of the Swedish law is clarified in its first paragraph:

§ 1 *The purpose of this Act is to facilitate the use of electronic signatures, through provisions regarding secure signature-creation devices, qualified certificates for electronic signatures, and the issuance of these certificates. The Act applies to certificate providers that are established in Sweden, and issue qualified certificates to the public.*

#### 3.2.1 Different types of electronic signatures

The Swedish Act on Qualified Electronic Signatures (SFS (2000:832), 2000) and the directive (the European Parliament, 1999) define three types of electronic signatures, whose greatest difference is their level of security. The electronic signature with the lowest level of security is simply called an electronic signature (or sometimes a weak or light electronic signature). The law defines electronic signatures as data in electronic form which is attached to or logically associated with other electronic data and is used as a method of authentication. This signature does not have to be coupled to a specific person. Thus, according to the legal definition, electronic signatures could, in practice, take the form of many different kinds of implementations, such as disposable codes or e-mails.

Definitions:

- **Signatory** – a person who is authorized to control a signature-creation device and acts either on his or her own behalf or on behalf of a natural or legal person or entity he or she represents
- **Signature-creation data** – unique data, such as codes or secret cryptographic keys, used to create an electronic signature
- **Signature-creation device** – software or hardware used to implement the signature-creation data
- **Secure signature-creation device** – see section 3.2.3
• **Signature verification data** – data, such as codes or open cryptographic keys, used to verify an electronic signature

• **Certificate** – an attestation in electronic form that links signature verification data to a signatory and confirms the signatory’s identity

• **Qualified certificate** – see section 3.2.4

• **Certificate provider** – a legal or natural person who issues certificates or who guarantees that the certificate of others complies with certain requirements

The next type of electronic signature, an *advanced electronic signature*, raises the bar for security as it is created using means that the signatory can keep under his or her exclusive control; thus it is linked exclusively to one signatory and can be used to identify the signatory. Furthermore, the advanced electronic signature is coupled to the data to which it relates in such way that any subsequent alteration of the data is detectable. This means that an advanced electronic signature guarantees the integrity of the data as well as the authenticity.

### 3.2.2 Qualified electronic signatures

The third, and by definition safest, type of electronic signature is the *qualified electronic signature* (sometimes called a *secure or strong digital signature*). This electronic signature meets all the criteria for an advanced electronic signature but is also based on a qualified certificate and created using a secure signature-creation device. The qualified electronic signature guarantees integrity, authenticity, confidentiality and nonrepudiation.

### 3.2.3 Secure signature-creation devices

The Act on Qualified Electronic Signatures defines a signature-creation device as software or hardware used to implement the signature-creation data. Furthermore, signature-creation data is defined as unique data, such as codes or secret cryptographic keys, used to create an electronic signature.

If a signature-creation device is to be declared secure, it must ensure that the signature is satisfactorily protected against forgery. Furthermore, the device has to ensure the following:

- The signature-creation data can, practically, occur only once.
- The signature-creation data cannot be derived by reasonable means.
- The signature-creation data can be satisfactorily protected by the legitimate signatory against use or access by others.

One final, and very important, requirement is also that a secure signature-creation device must never alter the data that is to be signed electronically or in any way prevent this data from being presented to the signatory prior to the signature process.

### 3.2.4 Qualified certificates

Qualified certificates are certificates as defined above, with the additional condition that they must be issued for a specific period of validity by a certificate provider that meets a number of requirements\(^5\). A qualified certificate must contain the following\(^6\):

\(^5\) These requirements are beyond the scope of this report §§ 9–12 Lag (2000:832) om kvalificerade elektroniska signaturer (Swedish Act on Qualified Electronic Signatures).
• An indication that the certificate has been issued as a qualified certificate
• The name and address of the certificate provider and information regarding in which state it is established
• The name or pseudonym of the signatory
• Special information regarding the signatory if relevant for the certificate’s purpose
• Signature verification data
• The certificate’s period of validity
• The certificate’s identity code
• The advanced electronic signature of the certificate provider or electronic signature with equivalent level of security
• Information regarding limitations of use

3.2.5 Legal validity of electronic signatures

In Sweden there is no particular legislation regarding contracts, deals or other agreements that have been established by electronic methods. These kinds of agreements or contracts are regulated by the same laws that are valid for traditional physical agreements or contracts.

One important aspect regarding Sweden’s legal system is the fact that it is based on the principle of free sifting of evidence. This free sifting of evidence means that there are no limitations on the sources of evidence one may plea to include in a legal process. The value of all presented evidence is analysed and determined by the courts and judges and is not predetermined by any laws. As a result of this principle, one could always request to include an electronic signature as evidence in legal processes that require such confirmation.

As a result of paragraph 17 in the Act on Qualified Electronic Signatures, qualified electronic signatures gain an important legal significance:

17 § If a requirement of a handwritten signature or its equivalent, contained in a law or regulation may be satisfied by electronic means, a qualified electronic signature shall be deemed to fulfil this requirement. However, in communication with or between government authorities, the use of electronic signatures may be subject to additional requirements.

The paragraph would have been crystal clear if it weren’t for the section saying:

If … [it] may be satisfied by electronic means …

This implies that the electronic signature’s ability to replace a handwritten physical signature has to be evaluated within every law that states a requirement regarding signatures.

But a clarification can be extracted from the Directive of the European Parliament and the Council on a Community framework for electronic signatures (the European Parliament, 1999), where it says:

6 §6 Lag (2000:832) om kvalificerade elektroniska signature (Swedish Act on Qualified Electronic Signatures).
7 English translation of “Fri bevisprövning”.
Member states shall ensure that an electronic signature is not denied legal effectiveness and admissibility as evidence in legal proceedings solely on the grounds that it is:

- in electronic form, or
- not based upon a qualified certificate, or
- not based upon a qualified certificate issued by an accredited certification-service provider, or
- not created by a secure signature-creation device.
Chapter 4 Technology

4.1 Smart cards

Smart cards are defined in terms of both physical characteristics, such as form factor and mechanical strength, and overall function, such as commands for card management or transmission protocols, in ISO/IEC 7816 (ISO, n.d.). ISO uses the term integrated circuit card (ICC) for the entity more commonly known by the public as a smart card. In this report the words ICC and smart card are used interchangeably.

The term smart card, or ICC, basically refers to two types of cards with embedded integrated circuit chip. The first and more basic one is somewhat a memory card. This memory card has no more smart functionality than any other “simple” storage device. These kinds of memory cards could be compared to USB storage devices and other similar hard drives. The data is accessed through a security module in the chip that protects the data from being rewritten or discarded in any unwanted way. One well-known example, known by almost all Swedish middle-aged citizens, is the outdated prepaid phone card, which was sold in practically every store. These prepaid phone cards were charged with a value corresponding to their price in Swedish currency (SEK) and were used in public phone booths to make phone calls. When a cardholder used the prepaid phone card to call someone, he or she could see the card’s value decreasing. The card was used until it was empty, and thereafter it was useless.

Next up is the more “intelligent” smart card, or ICC, with a built-in microcontroller, CPU\(^8\) or MCU\(^9\), with internal memory. These smart card chips function as a small computer that can perform quite advanced on-card operations, such as encryption and digital signatures (Alliance, 2002). Generally speaking, many associate the term smart card with these types of cards that actually have built-in microcontrollers (Ferrari, et al., 1998, p. 42). Hereafter, the term smart card will interchangeably be used to describe a smart card with CPU functionality or a smart card with a built-in microcontroller.

Smart cards can furthermore be divided into two major subcategories regarding how they communicate with a card reader. These two categories are contact smart cards and contactless smart cards.

---

\(^8\) Central processing unit.

\(^9\) Microcontroller.
Contact smart cards are usually inserted into a smart card reader, which communicates directly with the card via a conductive contact plate. In short, data transmission is conducted through physical contact points.

The physical contact point generally consists of a gold-plated chip with some connectors such as supply voltage (Vcc), clock signal (Clk), reset or input/output.

A contactless smart card chip communicates with a card reader via radio waves or remote contactless electromagnetic interface. There are three types of contactless smart cards, which differ in the operating distance between the card and the card reader (Smart Card Alliance, 2002). The three types of contactless smart card are defined in ISO 10536, ISO 14443 and ISO 15693. ISO 10536 defines close-coupling contactless cards, ISO 14443 defines proximity contactless cards and ISO 15693 defines vicinity contactless cards. Operating distances, the maximal distance between the card and the card reader, vary as follows:

- Close-coupling contactless cards – operating distance less than 2 mm
- Proximity contactless cards – operating distance up to 10 cm
- Vicinity contactless cards – operating distance up to 100 cm

The ISO/IEC for proximity and vicinity cards defines these abbreviations (Smart Card Alliance, 2002):

- PICC – Proximity integrated circuit(s) card
- PCD – Proximity coupling device
- VICC – Vicinity integrated circuit(s) card
- VCD – Vicinity coupling device

Simplified, PCDs are card readers for proximity contactless cards, and VCDs are card readers for vicinity contactless cards.

Combinations of these subcategories are also available. Two of these combinations are hybrid cards and dual-interface cards (Smart Card Alliance, u.d.). The hybrid card is equipped with two noninterconnected chips, one with a contact interface and the other with a contactless interface. The dual-interface card has both a contact interface and a contactless interface accessing the same single chip.

For the card reader to communicate with a contactless card, some kind of communication technique is required. This is true both for physical requirements, such as powering, and for communication protocols.

A technique called inductive coupling is used to fulfil the four requirements for a contactless card to communicate with a card reader:

1) Powering the card’s microprocessor
2) Clock signal transmission
3) Data transmission from the reader to the card
4) Data transmission from the card to the reader

The ISO standards specify that the power transfer to the PICC or the VICC is based on inductive coupling and that the power is transferred to the proximity card or the vicinity card using the frequency 13.56 MHz (Smart Card Alliance, 2002). Almost every smart card is operated passively. This means that power is transported from the terminal from the card (Rankl & Effing, 2010, p. 285).
The basic mechanism of inductive coupling is when an electrical current running through a coil creates a magnetic field that induces an electrical current in another coil (Rankl & Effing, 2010, p. 286).

**Figure 6 - Powering by inductive coupling**

The smart card and the coupling device, or terminal, use digital modulation methods to communicate. Transmitting data from the terminal to the smart card can be accomplished by all known digital modulation methods (Rankl & Effing, 2010, p. 286). Communication in the direction from the smart card to the coupling device is performed by load modulation, which is a type of amplitude modulation. Load modulation is accomplished when the smart card modulates, or changes, the voltage in the coupling device by changing its own load on the smart card coil (Rankl & Effing, 2010, p. 287).

**Figure 7 - Load modulation communication from smart card to card reader**

### 4.1.1 Data storage and data zones

As stated above, smart cards are data storage media. Most smart cards have hierarchically structured file management systems (Rankl & Effing, 2010, p. 421). The smart card file system is defined in ISO/IEC 7816-4. The standard specifies two types of files (Rankl & Effing, 2010, p. 423):

1) Directory files (called dedicated files, or DFs)
2) Data files (called elementary files, or EFs)

Directories, or DFs, can hold other directories and/or EFs in a hierarchical structure. The root directory file is called the master file, or MF. Every smart card must have an MF to be able to store files.

![Figure 8 - Smart card file structure](image)

Every file, both DFs and EFs, has a two-byte file identifier (FID) that is used to select the file. The EFs are furthermore divided into two types: the working elementary file (working EF) and the internal elementary file (internal EF). Data within the smart card that can be read or written by the external card terminal is stored in a working EF. Data intended for internal use within the smartcard, such as data for the operating system, source code or secret keys, is stored in internal EFs.

![Figure 9 - EFs and their accessibility](image)

A smart card file contains two parts, the file header and the file body. A smart card file’s access conditions and information are stored in the file header. All information about the file itself is stored within the file header. The access conditions rarely change and are specified when the file is created. This file header, or file descriptor, must contain at least the following (Rankl & Effing, 2010, pp. 437, 464):

- File name
- File type
- File structure
- File size
- Access conditions
The file body contains modifiable user data, which is linked to the file header by a pointer.

![File diagram]

*Figure 10 - A smart card file*

The file access conditions are divided into two types:

1) State-oriented file access conditions
2) Command-oriented file access conditions

State-oriented file access control is performed by comparing the current security state with the equivalent file access conditions. Command-oriented file access control defines what specific command or commands that have to be executed correctly to gain access to the file. Such commands are, in many cases, authentication or identification actions.

The current security state of a smart card can be one of two types depending on which part of the card it affects: local security state and global security state. The global security state affects the MF, which means it affects the whole smart card. The local security state is the security state of the currently selected DF.

The most important access commands for DFs are as follows (Rankl & Effing, 2010, p. 438):

- CREATE – create a new file
- DELETE – delete a file
- REGISTER – register a new file

Commonly used EF access commands are the following:

- INCREASE / DECREASE – calculations within a file
- INVALIDATE – block a file
- LOCK – permanently block a file
- READ / SEEK – read or search data in a file
- REHABILITATE – unblock a file
- WRITE / UPDATE – write data to file

It is the access control rules in the directory that govern whether files in the directory are allowed to be created or deleted (RSA Laboratories, 2000).

Files containing keys can be defined to perform the following operations (RSA Laboratories, 2000, p. 68):

- Compute checksum
- Compute signature
Verify checksum
Verify signature
Encipher
Decipher

The access control of a smart card allows for some logical security zones by defining the corresponding file access conditions (Australian Government, 2008). These logical security zones are as follows:

- Public zones
- PIN-protected zones
- Access-controlled zones
- Locked zones

These four logical security zones are defined by the file access condition categories defined by the ICC file structure standard and are included within the header of the files stored in the smart card (RSA Laboratories, 2000).

- ALW (public zone) – file access always possible
- CHV (PIN-protected zone) – file access possible after card holder verification, often by entering a PIN
- SYS (access-controlled zone) – file access permitted after system key presentation, typically available only to the card issuer
- NEV (locked zone) – file access never possible

The locked zone contains data that cannot be read, changed or deleted. This data, such as operating system and firmware, is created at the time of manufacture.

PIN-protected zones contain data that can be read, changed and/or deleted after the chip PIN code is successfully entered or some other valid authentication depending on the area of usage, such as biometrics, is provided.

In access-controlled zones, only authorized parties are allowed to read or change the information accessed. Such access-controlled zones can be accessed by users providing cryptographic keys. Multiple users can access different information within the access-controlled zone if different unique cryptographic keys are distributed for every user intended to access the zone.

The public zone contains information that can be read by anyone with physical access to the smart card and a corresponding card reader. This is nonprotected data.

Smart cards can also contain data physically printed on front and/or back of the card. One other potential read-only information holder on a smart card is a magnetic stripe.

![Figure 11 - Access control zones of a smart card](image-url)
4.1.2 Form factor

The most common smart card format is ID-1, defined in ISO 7810, with a format of 85.6 x 54 mm. Its thickness is 0.76 mm, and it has a corner radius of 3.18 mm (Rankl & Effing, 2010, p. 30). This standard goes back to the 1980s and was primarily intended for plastic cards with a magnetic stripe. The presence of a chip on an ID-1 card was defined several years later in other standards. If a card contains a microcontroller and conforms to the measurements of an ID-1 card, it is considered a smart card according to the ISO standard.

![Figure 12 - Dimensions of an ISO ID-1 smart card](image)

4.2 Authentication

"An identity credential is a tangible object, a piece of knowledge, or a sample of a person’s physical attributes that may be used during the process of confirming an individual’s claimed identity." (Smart Card Alliance, 2012)

Authentication processes differ significantly depending on the subject being examined, the purpose of the identification and the level of security required. For example, the standard methods of identifying a vehicle greatly differ from the methods of identifying the driver of the same vehicle. The strength of any authentication method is determined by the quality and diversity of its components. To ensure the highest integrity possible, the authentication method should include complementary mechanisms. Authentication methods are based on three fundamental authentication factors that confirm the identity of an entity, often a person, by verifying that the entity is what it claims to be:

- Ownership factor (something a person possesses, often a tangible object)
- Knowledge factor (something a person knows)
- Inherence factor (something characteristic about a person’s physiology and behaviour)

A piece of evidence provided to support an authentication factor is called an authentication token. Examples of authentication tokens include the following:

- Ownership factor – PKI asymmetric private key (digital certificate), smart card
- Knowledge factor – password or PIN
- Inherence factor – fingerprint or iris scan
When using more than one authentication factor while identifying a person or other entity, the level of security is strengthened. Three-factor authentication, which combines all of the above-listed factors, provides the highest level of security by strong authentication.

An authentication credential is an object that binds an identity to a token (Alliance, 2002, p. 6). A driver’s license is an identity credential that proves the qualification to drive a vehicle given to an individual by a license-issuing government.

Authentication tokens spreading across all authentication factors and usable in an EDL framework are passwords, digital certificates and biometrics.

4.2.1 Passwords

A commonly used but weak form of authentication is passwords. Passwords are often created by, or given to, and controlled by users, which means these authentication tokens have some drawbacks:

- The user or issuer might create an easily guessed password.
- The password will be nonsecret if the user shares it with others.
- The password can be written down in readable form so other people can read it.
- Users can reuse the same password across multiple systems.
- Someone can easily learn a user’s password by looking at the keyboard or display when the user inputs his or her password.

An authentication process based exclusively on password token authentication could be vulnerable to brute force attacks. Brute force attacks are launched on computer systems by trying combinations of words, often from a dictionary, until finding the correct combination of characters and digits. Weak password or short PIN authentication increases the systems vulnerability to brute force attacks. To prevent such brute force attacks, authentication systems often demand that users choose, or simply provide users with, passwords including lowercase and uppercase letters, digits and special symbols. The advantage of such passwords is that brute force attacks are much more complex and time consuming, but on the other hand, password complexity makes passwords harder to remember and more likely to be written down by users.

When using authentication processes like passwords or PINs, the authenticator system has to store these passwords or PINs in some form within the system to be able to verify the authentication tokens. Strong network and physical security is required to ensure that no one can break into the computer system and read the stored passwords. This risk can be reduced by using hash functions so the only data stored are hash values, or the output from hash functions where the original passwords are given as input. Should someone read the stored password data, only the hash values would be available.

An advantage of password based authentication is security at low cost. A disadvantage is that this is often security at low level. An advantage regarding ease of use is also that a password can easily be changed by a user or administrator, thus providing the administrator and user with ease of management. Passwords and PINs are widely used and accepted by many users throughout the world.

4.2.2 Public-key cryptography

Asymmetric cryptography, with a public- and private-key pair, guarantees that the entity tied to the private key, often by a digital certificate, indeed is who it claims to be.

Smart card technology provides a secure storage for private asymmetric keys within the cards. Private-key encryption can be performed securely on the card and cannot be revealed, exported or externally viewed (Alliance, 2002, p. 12). Smart cards can combine other authentication processes with the PKI
asymmetric cryptography approach. One example is designing a smart card as a physical identification card with a photograph of the holder and other readable information printed on the smart card.

4.2.3 Biometrics

Biometric factors have the advantage of binding an individual to a declared identity both during the identity credential registration process and during the authentication event. Therefore, biometrics is an important component of multifactor authentication (Alliance, 2002, p. 12).

Biometric authentication, or biometrics, uses physiological and behavioural characteristics for identification (Wayman, et al., 2005, p. 2). There are many examples of such physiological and behavioural characteristics such as fingerprints, palm prints, retina or iris scans, voice readings, shapes of body parts, optical skin reflection and even body odour.

The best biometric characteristics, providing the best possibility of good biometric authentication, have these five qualities (Wayman, et al., 2005, p. 3):

- Robustness – does not change over time
- Distinctiveness – shows great variation over the total population
- Availability – available in multiples across the entire population
- Accessibility – easy to scan or image by electronic sensor
- Acceptability – people do not mind the biometric characteristic

Biometric systems can be divided into two categories depending on which identification hypothesis they implement. If submitted samples are verified to being in the system, the hypothesis is called “positive identification”, and if the submitted samples are verified as not being in the system, the hypothesis is called “negative identification”. Where positive identification prevents multiple users from using one identity, negative identification prevents multiple identities for a single user. The system design can greatly vary depending on the choice of biometric identification category because positive identification allows for both centralized and decentralized storage of biometric comparison data but negative identification category requires centralized biometric comparison data storage. This means that positive biometric identification systems can store the comparison data on, for example, smart cards in the possession of users.

The choice of centralized or decentralized storage regarding positive identification depends on whether only one-to-one matching is performed. Such matching enables storage to be distributed across, for example, smart cards or magnetic stripe cards. Central storage could also be used in a one-to-one matching system to detect counterfeit cards or other attempts to manipulate comparison data and to be able to reissue lost cards or biometric data. To be efficient, systems performing one-to-N matching, where N is greater than one, have to have centrally stored comparison data (Wayman, et al., 2005, p. 13).

One drawback of using biometric factors is the possibility of lack of reliability due to misreading of biometric data.

4.3 Cryptography and PKI

Most data encryption techniques require two inputs, an algorithm and a key. The algorithm defines how the data is transformed into ciphertext, or encrypted data, and also how the ciphertext is transformed back into the original data. The key is an input to the algorithm that determines the functional output of the algorithm (Komar, 2008). The two major cryptography techniques are single-key cryptography and public-key cryptography.
Single-key cryptography, also known as symmetric-key encryption, is based on a single key used to both encrypt and decrypt the data. Some of the most well-known and widely used symmetric-key algorithms are AES, DES (3DES) and RC4, to mention a few.

![Symmetric key encryption diagram](image)

**Figure 13 - Symmetric-key encryption**

Public-key cryptography, also known as asymmetric encryption, uses a key pair, or two keys, one for encryption and the other for decryption. One of the keys is kept secret by the key holder and is therefore defined as the private key. The corresponding key can be distributed to the public and is defined as the public key. Data that is encrypted using the private or public key within a key pair can be decrypted only by using the other key within the same key pair. Public-key cryptography is utilized to perform digital signatures as well as user authentication (Choudhury, et al., 2002). Examples of asymmetric-key algorithms are RSA and DSA.

![Asymmetric-key encryption diagram](image)

**Figure 14 - Asymmetric-key encryption**

Many applications combine symmetric and asymmetric encryption to benefit from each method’s strengths (Komar, 2008, p. 9).

Public-key infrastructure (PKI) is a framework that provides hardware, software, policies and procedures to generate, manage and distribute keys and digital certificates. It is a framework for creating a secure method for exchanging information based on public-key cryptography. PKI is used to implement security and trust within the electronic world by defining secure methods for exchanging public-key cryptography–based data. Cryptography and PKI offer security functions such as confidentiality, nonrepudiation, authentication and integrity to fulfil security and trust requirements.

The PKI framework consists of the following components (Choudhury, et al., 2002, p. 30):

- Public- and private-key encryption technology
- Digital certificates
- Certification authority (CA)
• Registration authority (RA)
• Subjects or PKI clients
• Certificate repository

4.3.1 Digital certificates

A digital certificate associates a public key to an individual or another unique entity (Choudhury, et al., 2002). The digital certificate guarantees that the public key can be associated with the identified entity and that the same entity has access to and can use the corresponding private key. A digital certificate guarantees that the public key has not been modified or corrupted in any way and that the digital certificate actually represents the entity it is supposed to represent. A digital certificate may be compared with our physical ID cards or other identification documents and thus proves the identity of the owner. A simple example of a digital certificate is a public key signed by a trustworthy entity. For every such public key, there is a unique corresponding private key.

![Digital certificate](image)

A digital certificate is an electronically signed document in which a trustworthy entity, called certification authority (CA), confirms that a particular public key is associated with a certain person or entity. The individual or other unique entity is referred to as the certificate’s subject (Komar, 2008, p. 21). It is crucial that the CA validate the subject’s identity prior to issuing the digital certificate to ensure the reliability of the PKI.

A digital certificate contains a serial number, a digital signature by the CA, a public key belonging to the individual or other unique entity for whom the certificate is issued, the date of expiration and the name of the CA that has issued the digital certificate (Choudhury, et al., 2002, p. 31).

Using this digital certificate and CA, a receiver can verify who possesses the corresponding private key, whether the certificate – and thereby the signature – is valid, whether the certificate’s period of validity has expired and whether the certificate has been revoked. A receiver can also find out whether there are any limitations regarding the usage of the certificate. Certificates can be distributed by various technologies, such as online catalog services. Another common approach to distribution is for the signer to include his or her certificate, including the public key, together with the data that was digitally signed. Digital certificates are consequently a way to be able to securely distribute the subject’s public key to whoever interested.

Digital certificates, or public-key certificates, usually implement an international standard called X.509, which defines what information the certificate must and/or could include and in what format.
4.3.2 Certification authority (CA)

Issuers of digital certificates ensure that the certificates are valid by signing the contents during the process of creating them. A digital certificate is often created as a response to an entity’s request to obtain such a certificate. For the issuer, often a CA, to be able to associate a specific individual or other entity to a public key, and thereby create a valid and trustworthy digital certificate, this individual or other entity must be identified in a secure way. That such identification be thorough and the individual or entity’s identity be determined is crucial for maintaining the CA’s reputation and reliability.

A CA is a trustworthy party that signs a digital certificate with its private key and allows others to verify the digital certificate with the use of the corresponding public key. The CA both issues and maintains PKI digital certificates. One of the maintenance tasks for the CA is to handle the certificate revocation list, or CRL. This CRL is a list of all of the digital certificates issued by the CA that have been revoked for some other reason than expiration, and it has to be continually updated. CAs can be organized in a hierarchy (Utilize Windows, n.d.).

PKI architectures can be implemented in different ways regarding the number of CAs (or RAs; see the next section). Two of the most straightforward architectures are the single CA architecture and the hierarchical PKI architecture. The single CA architecture is the basic implementation, where one CA issues all certificates and distributes all CRLs to all PKI clients. The more common hierarchical PKI architecture arranges multiple CAs in a hierarchical structure, starting with the root CA as the top, or parent, node. The root CA issues certificates for the subordinate CAs beneath and indicates what kind of work the subordinate CAs are allowed to perform. The subordinate CAs can issue certificates to their subordinate CAs and PKI clients. Except for the root CA, every other CA has a single superior CA (Choudhury, et al., 2002, p. 40).

4.3.3 Registration authority (RA)

A registration authority (RA) is an intermediary that can be used between PKI clients and CAs. A CA can delegate the responsibility of verifying the subject’s identity to an RA. The RA collects client certificate requests, validates them and passes them on to a CA. The CA issues the digital certificate and returns the certificate to the subordinate RA. The RA acts as a verifier between PKI clients and a CA. The usage of RAs is suitable for scaling the PKI applications across different geographical locations (Choudhury, et al., 2002, p. 30).
4.3.4 Certificate repository

A certificate repository stores digital certificates and CRLs within a PKI architecture. Repositories are online, often publicly accessible databases where users can retrieve digital certificates and CRLs. The repositories are updated by the CAs when digital certificates are issued or revoked (Choudhury, et al., 2002, p. 67).

4.3.5 Revocation of digital certificates

Because a CA can issue digital certificates to other CAs and to PKI users and is responsible for validating that the subject has the correct credentials, it is also responsible for revoking digital certificates if they become prematurely invalid (Choudhury, et al., 2002, p. 61). A CA is responsible for indicating whether a certificate it has issued has been revoked (Cooper, et al., 2008, p. 7). A certificate may become prematurely invalid because the subject’s or CA’s private key has been compromised or because information regarding the subject stored within the certificate has changed and an updated certificate has to be issued. A subject can also be excluded from being a part of the PKI architecture (the subject is no longer certified by the CA issuing the subject’s certificate).

Revoked certificates are published in certificate revocation lists (CRLs). A CRL contains the serial numbers of the revoked certificates, and the list is signed by a CA. PKI users must ensure that a certificate used within the PKI architecture has not been revoked by a CA, to ensure the reliability of the architecture.

There is also a protocol useful in determining the current status of a digital certificate without the use of CRLs. This protocol, named online certificate status protocol, or OCSP, uses a request-response approach that informs the requestor whether the certificate is good, revoked or unknown (Myers M, 1999). The request consists of

- the protocol version,
- a service request,
- the target certificate identifier and
- (optional) extensions that may be processed by the OCSP responder.

The response consists of

- the target certificate identifier,
- the certificate status value (good, revoked or unknown),
- a response validity interval and
- (optional) extensions.

4.3.6 Format of an X.509 digital certificate

To facilitate the distribution and use of digital certificates, a universal international standard has been deployed, named X.509. Since the release of the first version in 1988, this standard has been developed and upgraded through versions 2 (1993) and 3 (1996). The different versions are backward compatible with each other because the newer versions are expansions of previous versions. The difference between version 1 and 2 meant only minor changes while more major changes can be seen while comparing version 2 and 3, when extension fields were included. Version 3 is almost exclusively in use today (Cooper, et al., 2008).

X.509 defines the contents allowed in a digital certificate, while the content is described using ASN.1. ASN.1, or Abstract Syntax Notation One, an International Standards Organization format (ISO format)
EDL – Electronic Driving Licenses

for the representation of data used to achieve interoperability between platforms. One major advantage of specifying a digital certificate in the ASN.1 format is that specific rules can be applied, among others the Distinguished Encoding Rules (DER), which transform the data into a format that is suitable for data transfer.

X.509 is supported by other protocols, such as PEM, PKCS, S-HTTP and SSL.
An X.509 digital certificate contains the following fields (Komar, 2008, p. 22) (Cooper, et al., 2008):

- Version number
- Serial number
- Certificate algorithm identifier
- Issuer name
- Validity period
- Subject name
- Subject public-key information
- Issuer unique identifier (only X.509 version 2 and 3)
- Subject unique identifier (only X.509 version 2 and 3)
- Extensions (only X.509 version 3)
- Certification authority’s digital signature (digital signature of fields above)

![Figure 18 - The format of an X.509 v3 digital certificate](image)

The ASN.1 syntax for an X.509 certificate (RFC5280)

```
Certificate ::= SEQUENCE {
```
tbsCertificate       TBSCertificate,
signatureAlgorithm   AlgorithmIdentifier,
signatureValue       BIT STRING 
}

TBSCertificate  ::=  SEQUENCE  {
  version         [0]  EXPLICIT Version DEFAULT v1,
serialNumber         CertificateSerialNumber,
signature            AlgorithmIdentifier,
issuer               Name,
validity             Validity,
subject              Name,
subjectPublicKeyInfo SubjectPublicKeyInfo,
issuerUniqueID  [1]  IMPLICIT
  UniqueIdentifier OPTIONAL,  
    -- If present, version MUST be v2 or v3
subjectUniqueID [2]  IMPLICIT
  UniqueIdentifier OPTIONAL,  
    -- If present, version MUST be v2 or v3
extensions      [3]  EXPLICIT Ext
  Extensions OPTIONAL     
    -- If present, version MUST be v3
}

Version (type EXPLICIT Version DEFAULT v1)
This integer indicates which X.509 version is used by the digital certificate. The integer is the version number minus 1 because of a zero-based index, so X.509 version 1 is defined by 0, version 2 by 1 and version 3 by 2.

If the fields issuerUniqueID or subjectUniqueID or both are in use, the version has to be set to at least version 2 (represented by the number 1). If the certificate contains any extension field, version 3 (represented by the number 2) is required. It is not forbidden to specify a higher version number than necessary even though it could be perceived as misleading.

Version number syntax
Version ::= INTEGER { v1(0), v2(1), v3(2) }

Version number example
Version: 3 (0x2)

Serial number (type CertificateSerialNumber)
This number, which must be a positive integer, is a unique number given to every certificate that is issued by a specific certification authority. This number, together with the CA’s identity, can therefore be used to uniquely identify a digital certificate. Because of the requirement of uniqueness, every CA has to be able to handle long serial numbers up to 20 octets.

Serial number syntax
CertificateSerialNumber ::= INTEGER

Serial number syntax
Serial Number: 267812 (0x41624)

**Signature** (type AlgorithmIdentifier)

This field contains an identifier that defines which algorithm is used by the issuer (CA) to sign the digital certificate. RFC3279, RFC4055 and RFC 4491 list supported algorithms. Other algorithms may also be supported. This field must contain the same algorithm identifier as the field signatureAlgorithm in the basic top sequence Certificate.

**Signature syntax**

AlgorithmIdentifier ::= SEQUENCE {
    algorithm OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL
}

**Signature syntax**

Signature Algorithm: sha1WithRSAEncryption

or

Signature Algorithm:

   Algorithm ObjectId: 1.2.840.113549.1.1.5 Sha1RSA
   Algorithm Parameters: 05 00

**Issuer name** (type Name)

The issuer field is defined as the X.501 type Name and identifies the entity that has issued and signed the certificate. This field must contain a nonempty distinguished name (DN). The X.501 type Name is an ASN.1 structure defined as follows (Cooper, et al., 2008, p. 19):

Name ::= CHOICE { -- only one possibility for now --
    rdnSequence RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET SIZE (1..MAX) OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
    type AttributeType, 
    value AttributeValue }

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY -- DEFINED BY AttributeType

DirectoryString ::= CHOICE {
    teletexString TeletexString (SIZE (1..MAX)),


Distinguished names are composed of attributes. Applications implementing PKI according to RFC5280 are required to recognize the following name strings (descriptors) but may also recognize others (Zeilenga, 2006) (Cooper, et al., 2008, p. 20):

<table>
<thead>
<tr>
<th>Must recognize</th>
<th>Should recognize</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>locality</td>
</tr>
<tr>
<td>organization</td>
<td>title</td>
</tr>
<tr>
<td>organizational unit</td>
<td>surname</td>
</tr>
<tr>
<td>distinguished name qualifier</td>
<td>given name</td>
</tr>
<tr>
<td>state or province name</td>
<td>initials</td>
</tr>
<tr>
<td>common name</td>
<td>pseudonym</td>
</tr>
<tr>
<td>serial number</td>
<td>generation qualifier</td>
</tr>
</tbody>
</table>

Common attribute types in X.500 format are as follows:

- **CN**  commonName
- **L**   localityName
- **ST**  stateOrProvinceName
- **O**   organizationName
- **OU**  organizationalUnitName
- **C**   countryName
- **STREET** streetAddress
- **DC**  domainComponent
- **UID** userId

**Issuer name example**

Issuer: C=SE, O=Posten Sverige AB, CN=Posten Sverige AB EID Mjukt CA v1

**Validity period** (type Validity)

The validity period defines a time interval within which the current certificate is considered valid. The interval is defined by a start time and date and an end time and date. The start and end time can be encoded as either UTCTime or GeneralizedTime. Applications must handle both time formats. In situations where no expiration date is set, the notAfter value should be set to the GeneralizedTime value of “99991231235959Z”. The digital certificate should not be considered valid if the time of control is not within this period of time.
UTCTime, or universal time type, is a standard ASN.1 type for representing date and time. The X.509 specification requires UTCTime to be expressed in the form YYMMDDHHmmSSZ, where YY is year, MM month, DD day, HH hour, mm minute, SS second, and Z Greenwich Mean Time. If YY ≥ 50, then the year is interpreted as 19YY. Otherwise, the year is interpreted as 20YY.

The X.509 digital certificate’s GeneralizedTime format, according to the X.509 specification, must use Greenwich Mean Time and include seconds (but not fractional seconds). The time format is YYYYMMDDHHmmSSZ, and the encoding of the letters has the same meaning. The GeneralizedTime format has the advantage of four digits that represent the year. Because of this, dates in year 2050 or later have to be encoded in the GeneralizedTime format.

**Validity period syntax**

```
Validity ::= SEQUENCE {
    notBefore      Time,
    notAfter       Time }
```

```
Time ::= CHOICE {
    utcTime        UTCTime,
    generalTime    GeneralizedTime }
```

**Validity period syntax**

Validity

Not Before: Nov 21 17:37:13 2003 GMT
Not After : Nov 21 17:37:13 2023 GMT

Validity (UTCTime format)

Not Before: 031121173713Z
Not After : 231121173713Z

Validity (GeneralizedTime format)

Not Before: 20031121173713Z
Not After : 20231121173713Z

**Subject name (type Name)**

The subject name field identifies the PKI user associated with the public key stored in the subject public key field. This field is defined as the X.501 type Name, the same as the issuer field. A CA can issue any number of certificates with the same subject name if they are issued to the same unique entity. Applications implementing the X.509 specifications must and should accept the same attribute types as the issuer field.

**Subject name example**

Subject: C=SE, CN=Mattias Häggström, S=Häggström, G=Carl Arvid Mattias, SN=198005261234

**Subject public key information**

10 “\xE5” = å, “\xF6” = ö
This includes the subjects, the PKI users, the public key and an algorithm field that indicates which algorithm the public key is to be used together with. The algorithm object identifier is the same as in the signature field.

**Subject public key information syntax**

SubjectPublicKeyInfo ::= SEQUENCE {
  Algorithm AlgorithmIdentifier,
  subjectPublicKey    BIT STRING
}

AlgorithmIdentifier ::= SEQUENCE {
  algorithm   OBJECT IDENTIFIER,
  parameters  ANY DEFINED BY algorithm OPTIONAL
}

**Subject public key information example**

Subject Public Key Info:

Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):

00:91:a7:5b:5e:dd:20:3d:8e:49:73:df:3c:80:e4:
fc:37:9d:c1:03:c0:f2:2e:ea:1e:ad:8e:18:94:36:
db:ce:09:3f:f4:c2:ef:2f:b5

Exponent: 65537 (0x10001)

**Issuer and subject unique identifier** (type IMPLICIT UniqueIdentifier OPTIONAL)

The optional issuer and subject unique identifier fields allow issuer and subject names to be reused over time. When one or both of these fields are present within a digital certificate, the version field has to be set to 1 or 2 (version 2 or 3).

**Issuer and subject unique identifier syntax**

UniqueIdentifier ::= BIT STRING

**Extensions** (type EXPLICIT Extensions OPTIONAL)

The extension fields make the attachment of extra information within the digital certificate possible. Issuers can choose to attach well-known extension fields or to define new ones, all depending on the area of usage of the digital certificate. An extension field contains three parts. These parts are an external
identifier, an indicator whether it is a critical extension or not and finally the extension data itself. The external identifier describes what type of data is contained within this extension field. Examples of this identifier could be binary, text, date or any other complex data structure. The indicator of critical extensions indicates whether it is acceptable to ignore this extension or not. A noncritical extension may be ignored by applications that do not use such an extension or do not recognize the extension. Applications should deny the use of digital certificates that contain critical extensions that the application does not recognize or cannot or does not process.

There are two types of extensions, constraint extensions and informational extensions (Gutmann, 2000). Constraint extensions limit the use of either certificates or the contents of certificates. Informational extensions contain additional information that can be used by the certificate users.

Some standard certificate extensions are the following:

Authority Key Identifier
Subject Key Identifier
Key Usage
Certificate Policies
Policy Mappings
Subject Alternative Name
Issuer Alternative Name
Subject Directory Attributes
Basic Constraints
Name Constraints
Policy Constraints
Extended Key Usage
CRL Distribution Points
Inhibit anyPolicy
Freshest CRL (a.k.a. Delta CRL Distribution Point)
Authority Information Access
Subject Information Access

Extensions syntax

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE {
  extnID OBJECT IDENTIFIER,
  critical BOOLEAN DEFAULT FALSE,
  extnValue OCTET STRING
    -- contains the DER encoding of an ASN.1 value
    -- corresponding to the extension type identified
    -- by extnID
}
Extensions example

X509v3 extensions:

X509v3 Basic Constraints:
  CA:FALSE
X509v3 Subject Key Identifier:
X509v3 Certificate Policies:
  Policy: 1.2.752.38.1.1.2.3.1.31
X509v3 Authority Key Identifier:
X509v3 Key Usage: critical
  Digital Signature, Key Encipherment

Signature algorithm (type AlgorithmIdentifier)

This field contains an identifier that defines which algorithm is used by the issuer (CA) to sign the digital certificate. RFC3279, RFC4055 and RFC 4491 list supported algorithms. Other algorithms may also be supported. This field must contain the same algorithm identifier as the field signature in the basic sequence tbsCertificate.

Signature algorithm syntax

AlgorithmIdentifier ::= SEQUENCE {
  algorithm OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm OPTIONAL
}

Signature syntax

Signature Algorithm: sha1WithRSAEncryption

Certification authority’s digital signature (signatureValue) (type BIT STRING)

This is the CA’s digital signature on the information contained within the digital certificate. This field contains the output of the CA’s signature algorithm.

Certification authority’s digital signature example

Signature Algorithm: sha1WithRSAEncryption

4.3.7 Digital signing

A digital signature is created by encrypting data, or most often a hash value of the data, with an asymmetric encryption algorithm. The encryption key used in this asymmetric encryption algorithm is the sender’s (signer’s) private key. The output of the encryption algorithm is called the digital signature (Choudhury, et al., 2002, p. 26).

Based on the technology of digital signatures, a signature scheme can actually be seen as three algorithms (Dent & Mitchell, 2004):

- An asymmetric key–generation algorithm that creates a private-key and a public-key pair. The private key is used for signing and the public key for verification.
- A signing algorithm. The signing algorithm inputs the private signing key and the data to be signed. The output is the digital signature of the input data.
- A verification algorithm. This algorithm inputs the public signing key, the digital signature and the data that was signed.

A signed message is created by using the data to be signed as the input to a hash function. The outputted hash value (digest) is used, together with the signer’s private key, as the input to the signing function. The signing function is an asymmetric encryption algorithm, and the output is the actual digital signature.

A signed message contains the following:

- The data to be signed
- Information regarding which hash function was used to calculate a hash value of the data to be signed
- The digital signature (i.e. private-key asymmetric-encrypted hash value of the data)
- Information that uniquely identifies the signer (identity of the CA issuing the signer’s digital certificate containing the signer’s public key and the digital certificate’s serial number)
- The signer’s digital certificate (containing the public key used for signing)
To verify digitally signed data, the following information is required:

- The signer’s public key
- The digital signature
- The signed data
- Information regarding which hash algorithm was used by the signer

To verify a signed message, a receiver first has to retrieve the signer’s public key. This is done either by extracting the attached public certificate or by retrieving the signer’s digital certificate from a certificate repository using the signer identification field. Decrypting the digital signature with the signer’s public key produces the hash value of the original data. Using the same hash function as the sender, the receiver inputs the sent data into the function and produces a hash value. Finally, the calculated hash value is compared with the decrypted hash value. If the two values are identical, then the digital signature is verified.

The receiver has to verify that the sender’s digital certificate is valid and has not been revoked if the digital signature is to be declared verified.

A verified digital signature and a valid signer’s certificate signed by a trusted CA identify the signer of the message (authentication) and ensure that the message has not been modified in any way (integrity). The sender also cannot claim to not have signed the data (nonrepudiation).
A digital countersignature, i.e. when more than one subject digitally signs a document, is accomplished by calculating a hash value using the received message’s verified digital signature as input, and digitally signing that digest. In other words, the countersigning is a digital signature of a digital signature. A countersignature is possible even without knowing the contents of a message (Nyström & Kaliski, 2000, p. 14).

4.3.8 PKI asymmetric data encryption

To guarantee confidentiality, the sender can use asymmetric encryption to ensure that only the intended recipient is able to read the sent data. The data to be sent is encrypted using a symmetric cryptographic function with a secret symmetric key. This secret symmetric key is then encrypted using an asymmetric cryptographic function together with the intended recipient’s asymmetric public key. The encrypted data, the information regarding which symmetric cryptographic algorithm was used, the public key–encrypted secret private key and some identification of the receiver, as well as the choice of the asymmetric encryption algorithm used, are all combined into an encrypted message sent to the intended receiver.

The use of symmetric encryption of the data and asymmetric encryption of the secret key is known as a hybrid cryptosystem.
4.3.9 Key lengths

The US government recommends a key length of at least 2,048 bits for RSA after 31 December 2010 (Barker, et al., 2009, p. 21). The length of the key affects the strength of the encryption. A short key would make it easier for someone to decrypt the encrypted data using brute force.

4.4 Vehicle technology

4.4.1 Controller area network (CAN) bus

CAN was developed in response to the growing need for an effective wiring system in road vehicles as a result of more and increasingly complex electronic devices.

Controller area network, or CAN, is a broadcast-type message-based protocol. A CAN network consists of a number of CAN nodes connected by a physical medium, the CAN bus. The broadcast type means that every CAN node in the network can hear all sent transmissions anytime. Therefore, local filtering is needed for every node to determine whether the sent transmissions are intended to be received by that node. Nodes thus react only to messages relevant for their own purposes.

Bosch developed the CAN protocol in 1983, and today there are many CAN chips available from many manufacturers. To standardize the CAN protocol, Bosch published CAN Specification 2.0 in 1991. In 1993 the International Organization for Standardization (ISO) published ISO 11898 as a complimentary standard to the CAN protocol. ISO 11898 has since been revised, from ISO 11898-1 to ISO 11898-6 in 2013.

ISO 11898 specifies that the CAN network uses a twisted-pair wire for communication purposes. The CAN network must consist of at least two nodes.

There are two kinds of CANs:
- High-speed CAN
- Low-speed CAN or fault-tolerant CAN

High-speed CAN is the most commonly used protocol throughout the vehicle industry when it comes to managing control devices such as engine control, brakes or emission systems. Low-speed CAN is used primarily for control of comfort devices such as seat or mirror adjustment.

The abovementioned local filtering, or frame acceptance filtering, is performed at the sublayer LLC (logic link control) within the data link layer (DLL) of the CAN and is performed by examining the relevance of the frames based on their identifier.

CAN bus communication messages can be one of four frame types. These frame types are data frames, remote frames, error frames and overload frames (Bosch, 1991).

Data frame

The data frame consists of seven fields and transports data from a transmitter to all receivers. The first field is a start-of-frame (SOF) field, which indicates the start of the data frame. The second field is the arbitration field, which has the main purpose of indicating the priority of data. This field contains an ID field which enables receivers to filter frames as mentioned earlier. The third field is the control field, which indicates the number of data bytes in frames to be transmitted. Next is the fourth field, the data field, which consists of the data to be transmitted within the data frame. Eight bytes is the maximum amount of data sent in a data frame. The fifth field of a data frame is the cyclic redundancy check (CRC) field, which checks the frame transmission error by a 15-bit CRC sequence based on a polynomial division. The sixth field, the ack field, is used by receivers to send acknowledgement to the transmitter.
and thereby indicate that the message was properly delivered. The seventh, and last, field of the data frame is the end-of-frame (EOF) field, which specifies the end of the current frame, and it consists of seven recessive bits.

<table>
<thead>
<tr>
<th>Start-of-frame</th>
<th>Arbitration Field</th>
<th>Control Field</th>
<th>Data Field</th>
<th>CRC Field</th>
<th>ACK Field</th>
<th>End-of-frame</th>
</tr>
</thead>
</table>

*Figure 22 - CAN data frame*

**Remote frame**

The CAN remote frame is sent by a receiver node to a transmitter node to signal that the receiver node requests transmission from the specified transmitter node. The remote frame has the same field layout as the data frame except that it lacks the data field included in the data frame.

<table>
<thead>
<tr>
<th>Start-of-frame</th>
<th>Arbitration Field</th>
<th>Control Field</th>
<th>CRC Field</th>
<th>ACK Field</th>
<th>End-of-frame</th>
</tr>
</thead>
</table>

*Figure 23 - CAN remote frame*

**Error frame**

The CAN error frame is transmitted by a node in case of a detected bus error, such as errors in data transmission, and it contains two fields. These two fields are the error flag and the error delimiter. A transmitted error frame reaches all other nodes in the CAN network. The protocol specifies that every node that receives an error frame must also broadcast an error frame to terminate current data transmission. For this reason the first error frame broadcasted has a primary-type error flag, and the responding error frames has the secondary-type error flags.

<table>
<thead>
<tr>
<th>Error flag</th>
<th>Error delimiter</th>
</tr>
</thead>
</table>

*Figure 24 - CAN error frame*

**Overload frame**

A CAN node that transmits an overload frame signals that it is not ready to receive any frames. This will result in an extra delay between sent data or remote frames. This frame consists of an overload flag and an overload delimiter.

<table>
<thead>
<tr>
<th>Overload flag</th>
<th>Overload delimiter</th>
</tr>
</thead>
</table>

*Figure 25 - CAN overload frame*
Chapter 5 Electronic driving license (EDL)

5.1 Merging information

Throughout Europe, and throughout many countries in the rest of the world, multiple identification and information documents are required and issued to citizens. Much of the information held within a document is duplicated across many other documents issued to the same holder. With the implementation of a European electronic driving license (EEDL), most of these documents could be merged into a single document. An EEDL could be used as a driving license, a passport, a medical card and an identity card. The possibilities are endless, and it would also be possible to combine the information from the mentioned documents with payment information. This report will not include any EDL design that includes information such as ATM card or credit card information. But even though such a design is beyond the scope of this report, merging these types of documents should be considered when deploying an EDL framework.

![Diagram of merging information into a single document](image)

*Figure 26 - Merging information into a single document*

5.2 Form factor and visible contents of European driving licenses

The EU issued Directive 2006/126/EC (the European Union, 2006) on driving licenses aiming to improve road safety and facilitate free movement of persons taking up residence in a member state other than the one issuing the license. The directive’s goal is to standardize driving licenses throughout the member countries of the EU. When the directive was written, over 110 different models of driving licenses existed within the member states and therefore created problems regarding administration and control of foreign driving licenses.

The Swedish Transport Agency\(^\text{11}\) is the central agency for issuing driving licenses in Sweden. As a result of Sweden’s membership in the EU, its government is obligated to implement the above directive, which was fulfilled by the introduction of the Swedish Transport Agency Statute (TSFS 2012:60) regarding the Swedish driving license layout and contents.

\(^{11}\) Transportstyrelsen.
The directive states (ANNEX I) (the European Union, 2006, p. 19) that a driving license within the member states of the EU must fulfil ISO 7810 (Identification cards – Physical characteristics) and ISO 7816-1 (Identification cards – Integrated circuit cards – Part 1 – Cards with contacts – Physical characteristics) regarding its physical characteristics. It is also concluded that driving licenses have to be made out of polycarbonate.

ISO/IEC 7816-1 specifies the physical characteristics of integrated circuit cards (smart cards or ICC) with contacts. It applies to ID-1 type identification cards and measures 85.60 x 53.98 mm.

The directive also determines that every implementation of driving licenses, according to the document, has to include some mandatory security features. The mandatory security features are as follows (ANNEX I) (the European Union, 2006, p. 19):

- UV dull card bodies
- Background pattern designed to be resistant to counterfeit
- Optical variable elements
- Laser engraving
- Some degree of security design background and photograph overlap

Additionally, at least three of the following security features have to be implemented: colour-shifting inks, thermochromic ink, custom holograms, variable laser images, ultraviolet fluorescent ink (visible and transparent), iridescent printing, digital watermark (in the background), infrared or phosphorescent pigments, tactile characters (or symbols or patterns).

The directive further states that a driving license shall have two sides with the word “Driving License” written on “page 1” in the language of the member state issuing the license (“Körkort” in Swedish), and optional is the name of the issuing member state. Page 1 shall also contain a distinguishing sign of the member state issuing the driving license (“S” for Sweden), printed in a blue rectangle and encircled by 12 yellow stars. The figures below show the layout and design of these driving licenses.
1. Surname  2. Other name(s)  3. Date and place of birth  4a. Date of issue  4b. Date of expiry  4c. Name of the issuing authority  4d. Reference number  5. License number  6. Photograph  7. Signature  8. Residence or postal address  9. Categories of vehicle(s) the holder is entitled to drive

Information enclosed in parenthesis is optional.

Swedish implementation – page 1 – in accordance with the Swedish Transport Agency Statute TSFS 2012:60 following European Union Directive 2006/126/EC

According to Article 1 in the EU directive on driving licenses, any member state may introduce a storage medium (“microchip”) as a part of the driving license on condition that EC-type approval can be granted (the European Union, 2006). Such approval can be provided given that satisfactory tamper resistance can be proven. The Swedish Transport Agency Statute (TSFS 2012:60) states in §7 that Swedish driving licenses shall have a reserved space allowing for the future use of microchip functionality.

12 See section 3.1 for explanation regarding categories of vehicles.
13 See appendix I (chapter 12.1) for additional information/restriction codes.
The EDL defined within the scope of this report aims to fulfil the requirements specified within the EU directive on driving licenses (the European Union, 2006) and to include additional information and functions that would further enhance usability and provide methods to securely implement a framework for EDLs.

The design of the EDL would preferably be a design that fulfils the specifications of the abovementioned EU directive, including smart card functionality. The EDL would preferably be implemented as a dual-interface smart card (see section 4.1), thus providing both physical (chip) and wireless access to the smart card. This would result in a visible design identical to the EU definition presented above, with the visible addition of a smart card chip on the back of the EDL.

Research in Sweden performed by Goldberg in the 1990s revealed that contact chip cards were not suitable, as a result of the contact chip being exposed to wear and tear, and therefore a new EDL system was developed using noncontact RFID cards (Goldberg, 1998). One solution would be to implement EDL on dual-interface cards that access the same information regardless of contact or contactless interface connection.

### 5.3 EDL information contents

In accordance with the EU directive (2006/126/EC) and to include additional information useful for society and the users, the EEDL should contain the following:

**Information in accordance with 2006/126/EC**

*Associations to previously presented definition figure are within parenthesis. This information is both printed on the face of the EDL and stored in electronic form on the EDL smart card.*

Issuing country (issuing state) (electronic representation in accordance with ISO 3166)

Surname (1)

Other name(s) (2)

Date of birth (3)

---

15 ISO 3166 – the International Standard for country codes and codes for their subdivisions.
Place of birth (3)
Date of issue (4a)
Date of expiry (4b)
Name of the issuing authority (4c)
Reference number (document number) (4d)
License number (identity) (5)
Photograph (image, picture) (6)
Signature (also stored as an image in digital form) (7)
Residence or postal address (8)
Categories of vehicle(s) the holder is entitled to drive (9)

Date of first issue (each category) (10)
Date of expiry (each category) (11)
Additional information/restriction(s) (each category) (12)

Additional information (only stored in electronic form on the EDL smart card)

Issuer information
Licensing country (ISO 3166)
Phone number of issuing authority
Web address of issuing authority
E-mail of issuing authority

Personal information
Personal identification number
Sex (gender)
Citizenship(s), nationality (ISO 3166-1 country codes, ISO 3166-2 state codes)
Language code(s) (preferred) (ISO 63917)
Height
Biometric information

Medical information
Organ donor
Blood type
Allergies
Identification number of the institution (European health insurance card)
Other medical information

16 In Sweden a unique personal identification number is used.
17 ISO 639 – the International Standard for language codes.
5.4 XML format – EEDL

The XML-based EDL, root element EDL, includes 10 child elements:

Document number
Issuing date
Expiry date
Issuing authority
ICE – In case of emergency
Identification – ID
Address
Driving license
Medical information
Traffic insurance

EDL is the root element and contains 10 child elements. EDL is a complexType element containing a sequence of the 10 elements. See Appendix I for a complete XML schema definition as well as an XML example of an EDL.

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema …>
  <xs:element name="EDL">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DocumentNumber" type="xs:positiveInteger"></xs:element>
        <xs:element name="IssuingDate" type="xs:date"></xs:element>
        <xs:element name="ExpiryDate" type="xs:date"></xs:element>
        <xs:element name="IssuingAuthority">
          <xs:complexType>
            …
          </xs:complexType>
        </xs:element>
        <xs:element name="ICE">
          <xs:complexType>
            …
          </xs:complexType>
        </xs:element>
        <xs:element name="Identification">
          <xs:complexType>
            …
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
5.4.1 Document number

A positive integer representing a unique identifier for every EDL issued.

5.4.2 Issuing date

A date, in the format YYYY-MM-DD\textsuperscript{18}, defining the date when the EDL was issued.

5.4.3 Expiry date

A date, in the format YYYY-MM-DD, defining when the validity of the EDL expires.

5.4.4 Issuing authority

A complexType that contains elements describing the authority issuing the EDL:

- Name
- Address (street, postal code, city, country\textsuperscript{19}, latitude and longitude)

\textsuperscript{18} Y=Year, M=Month, D=Day, for example 2014-11-13.
\textsuperscript{19} Y=Year, M=Month, D=Day, for example 2014-11-13.
- Telephone
- Web address
- E-mail address

### 5.4.5 ICE – In case of emergency

Describes whom to contact in case of an emergency regarding the holder of the EDL. The element ICE is a complexType holding two child elements, also complexTypes, representing a person and a company. The complexTypes person and company can be used in an EDL XML document to describe whom to call. Both the person and the company element contain string representations of name, telephone and e-mail.

### 5.4.6 Identification

The complexType Identification holds elements of data that identify the holder of the EDL, such as the following:

- Links to biometric data files stored in the smart card (photograph, handwritten signature, fingerprint, palm print, DNA, retina, iris)
- Personal identification number
- Surname
- Given names
- Gender
- Height (in cm without decimals)
- Date of birth (YYYY-MM-DD)
- Country of birth (country name and country code according to ISO 3166)
- Country of citizenship (country name and country code according to ISO 3166)
- 2nd country of citizenship (country name and country code according to ISO 3166)
- Preferred language (language name and language code according to ISO 639)

### 5.4.7 Address

A complexType entitled Address contains information regarding the driving license holder’s address:

- Street
- Postal code
- City country (country name and country code according to ISO 3166)
- Longitude and latitude, coordinates of address (double float format)

---

19 Both country name as a string and the country code according to ISO 3166.
5.4.8 Driving license

The complexType DrivingLicense contains elements that describe which vehicle types the license holder is permitted to drive. There are two child elements: the first is an integer storing the license number of the driving license. The second is a complex data type entitled categories consisting of up to 15 elements, also complexTypes, each named after the vehicle categories defined by the EU directive on driving licenses (the European Union, 2006). The elements, named after the vehicle categories, each contain the following data:

- Holder (a Boolean value, true or false, that indicates whether the license holder is permitted to drive vehicles of the specified vehicle category)
- Issuing date (YYYY-MM-DD, representing the date when the permission took effect)
- Expiry date (YYYY-MM-DD, representing the date when the permission expires)
- Restrictions regarding the issued driving license for the specified vehicle category defined by the name of the parent complex type element (see Appendix I on driving license restrictions)
- Additional information regarding the issued driving license category

A compressed XML schema definition of the complexType element DrivingLicense is presented below. Note that every child element of Categories contains the same five elements.

```xml
<xs:element name="DrivingLicense">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="LicenseNumber" type="xs:int"/>
      <xs:element name="Categories">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AM">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Holder" type="xs:boolean"/>
                  <xs:element name="IssuingDate" type="xs:date"/>
                  <xs:element name="ExpiryDate" type="xs:date"/>
                  <xs:element name="Restrictions" type="xs:double"/>
                  <xs:element name="AdditionalInfo" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="A1"/>
            ...
            <xs:element name="A2">
              ...
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      ...
      <xs:element name="DE"/>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
5.4.9 Medical information

The element MedicalInformation has the purpose of storing medical information regarding the license holder. The element contains child elements representing the following information categories:

- Organ donation
- European health insurance
- Medical conditions
- Allergies
- Medications
- Actions in case of emergency

The complexType OrganDonor has the purpose of replacing a physical organ/tissue donor card. The data type includes two Boolean values (true or false) that indicate whether the holder is willing to donate all or some organs and tissue. Furthermore, the element includes a string element for details about the holder’s wishes regarding organ donation. The OrganDonor element contains two dates (YYYY-MM-DD), start date and expiry date, which limit the validity of the license holder’s decision regarding organ/tissue donation.

The donation information could be signed by a third party, a witness, and this digital signature is stored in the digitalSignature field.

Organ/Tissue Donor Card

I ________________________ wish to donate:

Any needed organs and tissue

Only the following organs and tissue________________________________

Donor signature_____________Date_________

The following people have witnessed my commitment to be a donor:

Witness_______________________________

Witness_______________________________

“More than four out of five Europeans are in favour of the use of organ donation cards.”

“At the European level ... 12% of citizens have an organ donation card.”

(European Commission, 2007, pp. 15,20)

Figure 33 - Example of an organ/tissue donor card

The next complex element, EuropeanHealthInsurance, is included to replace a physical European health insurance card. This element includes the same information that is included on the face of the physical card. The contents of EuropeanHealthInsurance are as follows:

- Card identification number (integer)
- Issuing country (country name and country code according to ISO 3166)
- The insuring institution defined by child elements storing the identification number (integer) and name of the institution
• The personal identification number of the holder of the European health insurance card
• Issuing date and expiry date (YYYY-MM-DD) that limit the time of validity

The next child element of MedicalInformation is the string element MedicalConditions, which contains any medical conditions of the owner of the EDL. This field should, for obvious privacy reasons (see privacy chapter, section 9), be used to store only medical information that the holder wants to supply. The same privacy discussion should be made regarding the next string-defined element, named Allergies. This element has the purpose of storing information regarding the EDL holder’s potential allergies.

The next element, Medication, is a complexType consisting of three string elements that store information regarding medications taken by the EDL holder. The three parts of information are medication name, dosage and frequency.

The last element included in the MedicalInformation element is the string representation of the holder’s wishes regarding what actions should be taken in case of an emergency.

5.4.10 Traffic insurance

The complexType TrafficInsurance is meant to be used when the license holder has traffic insurance that is linked to the holder himself or herself and not to some vehicle. It consists of the following:

• An insurance identification number (integer)
• The name and address (see section 5.4.7, “Address”) of the insurance company issuing the traffic insurance
• Telephone number, web address and e-mail of the insurance company
• The start and expiry date (YYYY-MM-DD) limiting the validity of the traffic insurance

5.5 Storage and access control

Access to the EDL data has to be restricted and well organized to ensure that the information is read or written only by people or applications that are intended to do so. An EDL should have private and public data areas depending on the type of information stored and the field of usage. One example of usage is public access to medical data in case of an emergency. This could be information regarding the fact that the cardholder, for example, is epileptic or has severe allergies. Another case is when an EDL holder is involved in a traffic accident and wants to share his or her contact, insurance and/or other driving license information with some other driver or pedestrian involved in the accident.

Access control in an implemented EDL has to manage access by various individuals and/or applications ranging from the vehicle being driven, other drivers and passengers in case of an accident or police officers wanting to control the validity of the license and/or identity of the driver.

First of all, it is clear that some information on an EDL can be accessed without any access control. This is the information that is physically printed on the card. As this information is readable by anyone seeing the card within a reasonable range, the same information should be able to be accessed by anyone actually holding the card. Access could be restricted through the contactless interface so that unintentional card-to-reader communication is avoided.

The table below defines the levels of access control regarding the EDL’s data. Note that access control can differ between the contact and the contactless interface regarding the same data stored on a dual-interface smart card. Differences can be implemented regarding both reading and writing of data.
Access control regarding an EDL’s contents should be defined according to five levels:

- ALW – no access control
- CHV1 – cardholder verification 1 (used to verify cardholder for digital signatures)
- CHV2 – cardholder verification 2 (other cardholder verification than digital signatures)
- ADM – administrative authority access only
- NEV – no access allowed (only on-board access)

In addition, the card would also store

- CHV3 – personal unlocking code.

### 5.5.1 EDL key and certificate file storage and access

The EDL smart card should store cryptographic keys and digital certificates so that the EDL framework can utilize PKI features as a method of guaranteeing authentication, nonrepudiation, confidentiality and integrity. The EDL should include the following:

- Private keys (encryption and digital signature)
- Public keys (corresponding to the private keys)
- Digital certificates (trusted certificates)
- Secret keys (symmetric encryption)
- Other information (EDL information)

The EDL will be used by the holder to digitally sign an acknowledgement that he or she knows the action he or she is about to perform and knows about the related legislation. To enable such functionality in a smart card–based EDL framework, the smart card has to store a private key intended for digital signature purposes. This private key, intended for digital signatures, is protected from modifications, and its data is not readable. This private key can be used only for on-card digital signature operations; thus the private key never leaves the smart card and is never exposed. The digital signature operation utilizing this on-card-only digital signature private key can be performed only after cardholder verification. This cardholder verification (CHV1), often PIN, should be separate from cardholder verification enabling the cardholder to access other card holder verification (CHV2)–protected data files.

**CHV1** - PIN 1 - digital signature PIN - stored in EF\(_{chv1}\)

**CHV2** - PIN 2 - PIN - stored in EF\(_{chv2}\)

**CHV3** - PIN 3 - personal unblocking key - stored in EF\(_{chv3}\)

Read: NEV (not readable by anyone)

Update: ADM (can be updated by the card issuer or system administrator)

CHV1 and CHV2 could possibly be updated by the user after cardholder verification.

**Private key** (digital signature)

Key length: at least 2048 (RSA)

Read: NEV (not readable by anyone)

Update: ADM (can be updated by the card issuer or system administrator)

Compute signature: CHV1 (digital signature operation possible after cardholder verification)
In contrast to blocking access to the information stored within the private digital signature key file, the corresponding public key could be readable without any verification requirements. For security reasons it is, of course, required that the public key be distributed within, or at least verified using, a digital certificate prior to key usage. The digital certificate containing the signed public signature key is present within the smart card EDL so that the digital certificate can be appended to digitally signed data.

**Public key** (optional)
and
**Digital certificate** (including the public key corresponding to private signature key)
Read: ALW (readable by anyone)
Update: ADM (can be updated by the card issuer or system administrator)
Verify signature: CHV1
Stored in EF_{dc}

The user’s trusted digital certificates are stored in EFs and are readable, and could be used to verify digital signatures, by the user after CHV2 authentication. The trusted certificate data files can be updated only by the card issuer or the system administrator to guarantee security. These files include the trusted CA certificates.

**Trusted digital certificates** (including the public key corresponding to private signature keys)
Read: CHV2 (readable after cardholder verification)
Update: ADM (can be updated by the card issuer or system administrator)
Verify signature: CHV2
Stored in EF_{tdc1}, EF_{tdc2}, …, EF_{tdcx}

The EDL stores secret keys on the smart card to enable symmetric cryptography. The symmetric keys can be read, written or used to encrypt data by the user after CHV2.

**Secret keys** (used for symmetric encryption)
Read: CHV2 (readable after cardholder verification)
Update: CHV2 (can be updated after cardholder verification)
Encrypt: CHV2
Stored in EF_{sk1}, EF_{sk2}, …, EF_{skx}

Cardholder biometric verification (CHV_B) should securely store biometric data in files that can be read or modified only by the system administrators or EDL issuers.
5.5.2 EDL information file storage and access

Access to the license information could vary by type. Access control could differ depending on whether the access is requested through a contactless or physical interface. This is to prevent unwanted access to data.

A straightforward access control scheme would be to require CHV2 to access the EF containing the XML data of the EDL. Updates to this file could be made only by the issuing authority or another issuing administrator. This would mean the following:

**XML representation of EDL (stores driving license information)**

Read: CHV2 (readable after cardholder verification)

Update: SYS (can be updated by the card issuer or system administrator)

Stored in EF<sub>EDL</sub>

A more flexible data storage approach is to separate the XML representation of the EDL into smaller portions based on their information contents. This would make it possible to apply different access levels to different types of information stored within the EDL. As a result, every subcategory contained in separate files would have to be a standalone XML document.

<table>
<thead>
<tr>
<th>Contact interface</th>
<th>Contactless interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document number</td>
<td>ALW</td>
</tr>
<tr>
<td>Issuing date</td>
<td>ALW</td>
</tr>
<tr>
<td>Expiry date</td>
<td>ALW</td>
</tr>
<tr>
<td>Issuing authority</td>
<td>ALW</td>
</tr>
<tr>
<td>ICE</td>
<td>ALW</td>
</tr>
<tr>
<td>Identification</td>
<td>CHV2</td>
</tr>
<tr>
<td>Address</td>
<td>CHV2</td>
</tr>
<tr>
<td>Driving license</td>
<td>ALW</td>
</tr>
<tr>
<td>Medical information</td>
<td>ALW</td>
</tr>
<tr>
<td>Traffic insurance</td>
<td>ALW</td>
</tr>
</tbody>
</table>

*Figure 34 - Access control of EDL information*

This table explains that access controls are more widespread across the EDL information when the contactless interface is used. Someone with physical access to the EDL could extract the same information as someone simply looking at the face of the card. The contactless interface access rights make sure that unwanted remote access is prohibited.

The issuing authority or some other approved entities are the only ones approved to make changes to the information contained within the EDL. The user is authorized to update personal and public information not vital to the issued permissions to drive vehicles.

This division of information would also result in information subsets of the EDL being easier to be digitally signed separately. For example, a subset of the medical information, such as organ donor information, could be signed by a witness.
5.6 Digital certificate extensions

Information regarding the license holder’s EDL information could also be stored as extensions in the digital certificate of the holder. This would be an informational extension (see section 4.3.6) containing information used by the certificate user. Every certificate extension has to be identified by an object identifier (OID). This OID, representing the EDL information extension, should be combined with the noncritical extension attribute, indicating that this informational extension is not vital to the use of the digital certificate.

An example showing a certificate extension of a driver license regarding vehicle categories AM, A and B can look like the following:

```
Extension ::= SEQUENCE {
    extnID      XXX.YYY.ZZZ (OID representing the EDL extension)
    critical    FALSE,
    extnValue   AM, A, B
}
```

This approach would result in a digital certificate issued and signed by a trusted CA, including the EDL extension data.

But a drawback of including driving license information in digital certificate extensions is that digital certificates are often publicly accessible. Someone could easily extract driving license information from digital certificates containing EDL information. Digital certificates should not contain data that has to be kept secret at any point, and in some cases, perhaps issued driving license information should not be published to the public.
Chapter 6 Driver and vehicle verification service (DaV-VS)

6.1 Verification system

A verification system is an online resource that inputs a driver’s identity and the identity of the vehicle to retrieve the following:

- Vehicle category
- Vehicle status
- Driving license category
- Driving license status

The vehicle category can be any of the vehicle categories defined by the EU standard presented in section 3.1 (the European Union, 2006), and the status says whether the vehicle is allowed to be driven in terms of, for example, taxes, insurance or vehicle inspections.

The driving license category includes the categories on the driver’s EDL, which are the same as the vehicle categories defined above. The driving license status communicates the status of the driver’s license and might state, for example, that the driving license has been revoked, temporarily suspended or reported as lost.

The driving license verification system would also have to handle driving license restrictions (see Appendix I) regarding the vehicle about to be driven. Verifying that the driver follows the regulations is more difficult for some restrictions than others. For example, vehicle restrictions regarding vehicle equipment are easier to verify using stored vehicle data than whether the user is wearing glasses or contact lenses. Of course, vehicles can be modified so they no longer match the stored data standards, but this requires more work than the driver simply taking off his or her glasses.

6.2 Driver and vehicle database (DaV-DB)

To be retrieved by a verification system service, driving license information and vehicle information data should be stored within a database. This driver and vehicle database (DaV-DB) could be stored either within the verification system, as part of the verification system itself, or at a remote location.

Most countries today already store records of their issued driving licenses in electronic form. These databases or other storage systems are used to perform queries regarding driving license data and vehicle information. These systems could hopefully be used as DaV-DBs providing the verification system with relevant information.

The definitions presented below are a simplified representation of databases storing basic driver and vehicle information.

Definition of vehicle categories

<table>
<thead>
<tr>
<th>AM</th>
<th>A1</th>
<th>A2</th>
<th>A</th>
<th>B</th>
<th>BE</th>
<th>C1</th>
<th>C</th>
<th>C1E</th>
<th>CE</th>
<th>D1</th>
<th>D</th>
<th>D1E</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>
Database containing vehicle information

<table>
<thead>
<tr>
<th>VIN</th>
<th>Plate</th>
<th>Cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEF12345</td>
<td>ABC123</td>
<td>4</td>
</tr>
<tr>
<td>FEDCBA54321</td>
<td>XYZ321</td>
<td>7</td>
</tr>
</tbody>
</table>

Database containing driver information (example of driver issued EDL category B)

<table>
<thead>
<tr>
<th>PIDnr*</th>
<th>AM</th>
<th>A1</th>
<th>A2</th>
<th>A</th>
<th>B</th>
<th>BE</th>
<th>C</th>
<th>C1</th>
<th>C1E</th>
<th>CE</th>
<th>D</th>
<th>D1</th>
<th>D1E</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYYYMMDD8565</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

* Personal identification number

The databases should contain additional information:

- Modification history
- Issuing information
- Issuing and expiry dates
- Restrictions or other additional information
- Addresses

The countries implementing an EDL framework should consider merging their national DaV-DBs into one international database containing all information regarding vehicles and driving licenses. This database could be redundantly stored, as a local copy, in every member nation. This would, though, pose a risk to privacy (see chapter 9) if the information were to be misused by some person disrespecting personal privacy.
Chapter 7 Electronic driver license (EDL) verification

Every implementation of a framework for EDLs depends on some mandatory parts to ensure full validity:

- Driver authentication
- Vehicle authentication

Such driver and vehicle authentication is required to be able to perform driver’s license verification. These are mandatory steps to be able to guarantee that a person cannot drive a vehicle that he or she has not been issued a proper license to drive. An EDL framework could implement one or more of these parts, resulting in frameworks with various levels of security. Each step can contain various levels of security, and the total level of security of a complete framework depends on the validation steps that are included and each step’s level of security.

The level of security often correlates to the cost to implement and deploy the system. The level of security within each step is also dependent on the design of the EDL. Are the issued EDLs stored on local storage devices, or are they stored exclusively in a remote database? Some EDL frameworks could be implemented by incorporating electronic information into today’s existing driving licenses, which would mean a combination of physical layout and security as well as digital design. The combinations of possibilities are almost endless, and they all have advantages and disadvantages depending on such previously mentioned areas as security, deployment and maintenance costs and user acceptance. One of the most complicated aspects of the deployment of a complete, fully functional and highly secure EDL framework is the legal aspects and also questions regarding license holder privacy.

The vehicle to be driven has an on-board EDL verification system, connected to the vehicle’s high-speed CAN that has to verify that the current driver has a valid driving license for the vehicle category. This verification differs depending on the approach regarding the connection to the driving license verification database or verification of local data. This control mechanism can be performed in an online or offline mode.

Local storage of driver license information would be

- in the driver’s EDL,
- in a database, or other digital storage medium, stored within the vehicle or
- in a database stored separately from the vehicle but physically connected to the vehicle.

Remote or online storage of driving license information would be

- in a database, or other digital storage medium, that is not stored within the vehicle computer system and that is not physically connected to the vehicle.

In addition, the system design also depends on the type of vehicle network connection available, which could be

- permanent,
- partial, irregular and nonpermanent,
- regularly synchronized (daily, monthly or yearly) and/or
- offline.


7.1 Driver authentication

As in traditional, nonelectronic driving license contexts, the license holder has to be securely identified prior to being approved for an EDL. That this initial identification be done properly is vital for maintaining reliable authentication all through the complete EDL framework.

Most countries and governments, or other driving license–issuing authorities, demand that a license holder prove himself or herself qualified to drive a vehicle prior to being issued a driving license for that vehicle category. The actual driving tests are beyond the scope of this report, but it is crucial that the person performing the driving test be securely identified. Such initial identification can be done by traditional authentication measures, such as presenting the driving instructor with a valid passport or photo ID or having another securely identified person who knows the driving license applicant guarantee that the applicant is who he or she claims to be.

This initial identification could be performed by electronic authentication methods given that citizens are issued valid electronic identification data prior to being issued an EDL. But physical identification has to be made at some point in order to issue an identification document. This initial physical authentication is often done by parents or a government official when a citizen is issued his or her first identification document. Some countries issue electronic authentication documents containing public key information to their citizens at an early age, as with the Malaysian MyKad or MyKid.

When an initial electronic authentication document is issued, it can be used to authenticate the person for him or her to be able to obtain future identification documents.

The design of the EDL and the included authentication factors affect the information that has to be collected from a driver being issued an EDL.

7.1.1 Password

One of the simplest forms of driver authentication is the driver entering a password or PIN into a display in the vehicle that he or she is about to drive. A password, depending on its complexity, guarantees variable but often a low level of security. This knowledge authentication factor, when used without other authentication factors, could be a way of preventing unwanted access to a vehicle by anyone not knowing the password. A password could be a part of a larger EDL framework but could probably not qualify as a valid standalone EDL framework.

Password-based authentication requires that the entered password at some point be compared to a stored password to validate the input. This stored password could, in its simplest form, be stored locally, unencrypted, someplace within the vehicle’s computer, not linked to any user data. This authentication would have the same functionality as a key. The authentication prohibits anyone from starting the vehicle without knowing the correct password or PIN. If the password is given to, or created by, a specific person and somehow linked to that specific person, it could serve as a form of authentication for the person. A vehicle password would not function as driver authentication in any other way than restricting access to that vehicle.

The cardholder verification (CHV) of a smart card EDL uses a stored password for authentication purposes. This password-based authentication has the advantage of safe storage of the password inside the smart card accessible only by EDL issuers and the smart card’s internal functions.

7.1.2 Biometrics

A more advanced form of driver authentication is the use of biometric authentication. When a driver requests to start a vehicle, he or she has to be identified by biometric data. Such data could, for example, be fingerprint(s), palm print(s), retina scan, weight and body measures or voice recognition. Biometric
driver authentication requires that comparison data be stored either locally or remotely, associated to some unique individual data, to perform valid and reliable authentication.

As with the password CHV of an issued smart card EDL, biometrics could be used to authenticate the driver and access EDL smart card data.

### 7.1.3 PKI

Public-key infrastructure (PKI)-based authentication based on private- and public-key asymmetric encryption provides a secure method for identifying a driver. If the private key is not compromised and the driver’s public key is securely linked to the unique individual, as a PKI digital certificate, a secure and reliable authentication is possible.

The risk of PKI authentication is that the driver could voluntarily give private-key access to another person. This voluntary choice to compromise the user’s private key could be prevented by protecting the key on a smart card with biometric-only access.

The private key of the driver should always be protected by a password, a PIN or biometrics.

### 7.2 Vehicle authentication

A vehicle is, as of 1981, identified by its vehicle identification number (VIN) (National Highway Traffic Safety Administration, 2014). The VIN is defined in ISO 3833 and ISO 3780 and consists of 17 characters and letters. The VIN was first used in 1954, in the United States, but was not standardized until 1981 in ISO 3833. Similar to biometric information for a person, like a fingerprint, a VIN uniquely identifies a specific vehicle. The VIN can be found in multiple places in a vehicle, and complete alteration of all VINs in a vehicle is a complex and difficult enterprise.

The VIN consists of 17 characters and letters. The letters I, O and Q are not used in a VIN (Insurance Bureau of Canada, 2014).

![Vehicle Identification Number (VIN)](image)

Another, less secure authentication of a vehicle is by inspecting the vehicle’s license plate. License plate characteristics vary between countries, and license plate modification is a lot easier than manipulating VINs.

Each vehicle included in an EDL framework should be issued a digital certificate containing the VIN and the license plate number within the subject field. The digital certificate subject field should include at least the VIN (serial number OID) but if possible also the license plate number (common name OID) and the country code of registration (country OID) for authentication purposes.

Subject: country=SE, commonName="ABC123", serialNr=XXXXXXXXXXXXXXXXXX (17 digit VIN)
Other attributes could also be used, such as the owner OID (in the subject), to define the identity of the owner (distinguishedName). The subject name fields could also be used to define the type of vehicle that is the subject. For example, a name OID could contain a text string like “Volvo XC70” to indicate that the vehicle with the given VIN is a four-wheel-drive Volvo. The extra information regarding vehicle information can often be extracted from the VIN, which means that such information is nonvital for an EDL framework.

The vehicle’s private key should be stored in nonaccessible memory in the vehicle’s computer. This private key should be used only in an encryption function by which one could, for authentication reasons, input data and receive signed data as output. Output data is signed with the vehicle’s private key.

Vehicle authentication could be accomplished by inputting a timestamp into the vehicle’s computer and as receiving a timestamp signed by the vehicle’s private key as output. This signed, asymmetric-encrypted timestamp is then decrypted using the vehicle’s public key, extracted from a valid nonrevoked digital certificate issued by a trusted CA to that vehicle. If the decrypted timestamp matches the input timestamp, then the vehicle authentication is a success.

The VIN can also be retrieved by accessing the OBD of vehicles. A security risk becomes apparent as vehicle software and hardware, such as OBD readers and writers, become more advanced and are possibly able to manipulate the digitally stored VIN of a vehicle. An EDL framework is dependent on the VIN of a vehicle, its fingerprint, being impossible to change.

**7.3 Online EDL verification**

In a perfect world, license verification should be performed in real time with every vehicle connected to an online, always-updated license verification database. In these online verification systems, the EDL reader has to retrieve only the unique identifier for the person about to drive the vehicle. This unique identifier is then sent, by uplink, to the license verification database, which returns the driver’s license categories of the driver. As an alternative, the vehicle sends the unique identifier of the driver and also the VIN to the database. In this case, the database could return a return code indicating whether the driver is permitted to drive the current vehicle. The message to the license verification database should be digitally signed by the driver. The return code (could be unique for every unique vehicle) could be encrypted with the vehicle’s public key so that only the specified vehicle, with the corresponding VIN, can open the message containing the start code.

This online verification scheme requires a very limited amount of information stored on the EDL smart card. In theory, the only information required on the card is a digital certificate containing the driver’s unique identifier so that he or she can be identified in a secure way. An extension of this verification scheme would be a system that identifies a driver through fingerprints, retina scan or facial recognition (or similar identification technology), therefore removing the need for any smart card solution.

One major drawback of this online approach is the never-ending demand for online connectivity. If a vehicle lacks connection to the license verification database, the vehicle will not get a response message and therefore will not start. Failed connectivity could be a result of broken hardware or lack of Internet coverage, to mention two of many possibilities. This online verification approach demands total connectivity every time someone is about to start a vehicle, and is thus not yet appropriate for real-world implementation.
7.3.1 Permanent network connection

If every vehicle had permanent network connectivity, the deployment of an EDL framework could take great advantage of today’s technology regarding electronic identification. The driving license information for every license holder could be stored in secure databases accessed by vehicles after their drivers are identified on requesting to drive such vehicles. Secure identification of drivers could utilize today’s electronic identification (eID) methods, often based on digital certificates issued by trusted CAs. The driver would only have to present his or her identity to the verification service, and every individual’s driving license information would be accessible at all times. This eID-only approach can also be appropriate in scenarios when it is acceptable that vehicles loosing network connectivity cannot be started.

To prevent tampering with vehicle data, every vehicle should also identify itself using a digital certificate issued to every vehicle included in the EDL framework. Vehicle information, as well as driving license information for drivers, could be stored in secure databases accessed prior to accepting start-up of the vehicle. This section describes the sequence of this PKI-based online EDL verification.

![Diagram of online EDL verification](image)

**Figure 36 - Online EDL verification**

Creating, signing and sending a DRIVE_REQUEST

Driver \( d_1 \) enters vehicle \( v_1 \).

\( v_1 \) has VIN \( v_{1\text{vin}} \), license plate number \( v_{1\text{plate}} \) and vehicle category \( v_{1\text{cat}} \).

\( d_1 \) has a valid EDL for \( v_{1\text{cat}} \).

\( d_1 \) pushes the start button (or turns the ignition key).

\( v_1 \) detects that an unknown driver \( d_1 \) has requested to start and drive \( v_1 \).

\( v_1 \) creates DRIVE_REQUEST \( d_{1_1} \) containing timestamp \( t_1 \), a random sequence number \( x_1 \) and the vehicle’s ID consisting of the license plate number \( v_{1\text{plate}} \text{presented} \) and VIN \( v_{1\text{vin}} \text{presented} \). \( v_1 \) has to ensure that \( v_{1\text{plate presented}} = v_{1\text{plate}} \) and \( v_{1\text{vin presented}} = v_{1\text{vin}} \) to guarantee full security.
DRIVE_REQUEST = [t₁, VEHICLE_ID, x₁]

VEHICLE_ID = [v₁plate_presented, v₁vin_presented]

Figure 37 - DRIVE_REQUEST containing VEHICLE_ID

v₁ presents the vehicle’s ID (v₁plate_presented or v₁vin_presented or both) and the timestamp of dr₁ to d₁, requiring d₁ to digitally sign the information as a verification that d₁ wishes to start v₁.

For example:

Please confirm that you, at local time YYYY-MM-DD HH:MM:SS, would like to start vehicle with license plate number ABC123 and vehicle identification number (VIN) ABCDEF123456.

v₁ prompts d₁ to connect an EDL to the vehicle’s EDL reader.

To confirm, please connect your EDL and digitally sign the request.

d₁ connects EDL₁ to the EDL reader of v₁.

EDL connection detected.

The EDL reader of v₁ prompts d₁ to identify himself or herself by password, PIN or biometrics (CHV1) to be able to access the private key of d₁ (d₁private) securely stored within EDL₁ in the EFpk.

Enter PIN or press finger against scanning device.

The EDL reader of v₁ signs the DRIVE_REQUEST with d₁private, creating a SIGNED_DRIVE_REQUEST.

SIGNED_DRIVE_REQUEST = signed(DRIVE_REQUEST, d₁private).

The SIGNED_DRIVE_REQUEST contains the DRIVE_REQUEST, the signer’s ID, the CA’s ID, a hash function and the signer’s digital certificate.

Figure 38 - SIGNED_DRIVE_REQUEST

The SIGNED_DRIVE_REQUEST is returned to the vehicle.

Request signed, thank you.

If the vehicle stores a local permission registry indicating who is allowed to drive the vehicle at a given moment in time, the two next steps in italics are included:

v₁ verifies the SIGNED_DRIVE_REQUEST to authenticate d₁. v₁ now extracts d₁’s ID d₁id.
$v_1$ compares $d_{id}$ to its local permission registry to control whether $v_1$ (by its permitted administrators) allows $d_1$ to start $v_1$, based on driver identification, the owner’s rules and the local time. Starts the localPermissionSequence($\text{driverID}$), which returns true or false.

$v_1$ countersigns the SIGNED_DRIVE_REQUEST, creating a COUNTERSIGNED_DRIVE_REQUEST:

\[
\text{COUNTERSIGNED\_DRIVE\_REQUEST} = \text{sign}(\text{SIGNED\_DRIVE\_REQUEST}, v_{\text{private}})
\]

This COUNTERSIGNED_DRIVE_REQUEST serves as proof that $v_1$ has locally approved the DRIVE_REQUEST approved by $d_1$. Both $d_1$ and $v_1$ have approved that $d_1$, if he or she has the correct driving license for the vehicle type, is permitted to drive the vehicle.

![Figure 39 - COUNTERSIGNED\_DRIVE\_REQUEST](image)

$v_1$ sends the COUNTERSIGNED_DRIVE_REQUEST to the driver and vehicle verifying service DaV-VS.

Sending request...
Request sent... waiting for response.

**Verifying and extracting information from a DRIVE_REQUEST and returning a signed DRIVE_RESPONSE**

DaV-VS verifies $v_1$’s signature of the COUNTERSIGNED_DRIVE_REQUEST using $v_1$’s public key $v_{\text{public}}$ to identify $v_1$ and extracts $v_{\text{vin}}$ and $v_{\text{plate}}$ (from the identification information guaranteed by the digital signature) and the SIGNED_DRIVE_REQUEST.

DaV-VS verifies $d_1$’s signature of the SIGNED_DRIVE_REQUEST using $d_1$’s public key $d_{\text{public}}$ to identify $d_1$ and extracts $d_{id}$ (from the identification information guaranteed by the digital signature) and the DRIVE_REQUEST.

DaV-VS extracts $v_{\text{plate}}_{\text{presented}}$ and $v_{\text{vin}}_{\text{presented}}$, $x_f$ and $t_f$ from the DRIVE_REQUEST.

DaV-VS compares if $v_{\text{plate}}_{\text{presented}} = v_{\text{plate}}$ and $v_{\text{vin}}_{\text{presented}} = v_{\text{vin}}$ to determine that $d_1$ signed presented vehicle identification data corresponding to the identified $v_1$.

DaV-VS checks that the timestamp of the DRIVE_REQUEST is within the allowed timespan, lower boundary $\leq t_f$ $\leq$ upper boundary.

DaV-VS sends a VEHICLE_STATUS_REQUEST to the driver and vehicle database DaV-DB. The VEHICLE_STATUS_REQUEST contains the random sequence number $x_f$ received in the DRIVE_REQUEST, as well as vehicle identification information $v_{\text{plate}}$ and $v_{\text{vin}}$.

\[
\text{VEHICLE\_STATUS\_REQUEST} = [x_f, v_{\text{plate}}, v_{\text{vin}}]
\]

DaV-DB checks its records of the current vehicle and creates a VEHICLE_STATUS_RESPONSE containing the information regarding the vehicle status, the vehicle category, the sequence number of the request, $v_{\text{plate}}$ and $v_{\text{vin}}$. DaV-DB returns the VEHICLE_STATUS_RESPONSE to DaV-VS.

\[
\text{VEHICLE\_STATUS\_RESPONSE} = [x_f, v_{\text{plate}}, v_{\text{vin}}, v_{\text{status}}, v_{\text{category}}]
\]
### Vehicle status (v\text{status})

<table>
<thead>
<tr>
<th>Status</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>0</td>
</tr>
<tr>
<td>Uninsured</td>
<td>1</td>
</tr>
<tr>
<td>Unregistered</td>
<td>2</td>
</tr>
<tr>
<td>Stolen</td>
<td>3</td>
</tr>
<tr>
<td>Not in traffic</td>
<td>4</td>
</tr>
<tr>
<td>Driving ban</td>
<td>5</td>
</tr>
<tr>
<td>Vehicle tax not paid</td>
<td>6</td>
</tr>
<tr>
<td>Vehicle not found</td>
<td>7</td>
</tr>
<tr>
<td>Error</td>
<td>8</td>
</tr>
</tbody>
</table>

### Vehicle category (v\text{category})

<table>
<thead>
<tr>
<th>Category</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>0</td>
</tr>
<tr>
<td>A1</td>
<td>1</td>
</tr>
<tr>
<td>A2</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B1</td>
<td>4</td>
</tr>
<tr>
<td>BE</td>
<td>5</td>
</tr>
<tr>
<td>C1</td>
<td>6</td>
</tr>
<tr>
<td>CE</td>
<td>7</td>
</tr>
<tr>
<td>D1</td>
<td>8</td>
</tr>
<tr>
<td>DE</td>
<td>9</td>
</tr>
<tr>
<td>Error</td>
<td>10</td>
</tr>
</tbody>
</table>

DaV-VS sends a DRIVER\_STATUS\_REQUEST to DaV-DB. The driver status request contains the sequence number \(x_1\) of the DRIVE\_REQUEST, the driver’s identity \(d_{lid}\) and the vehicle category \(v_{lc\text{ategory}}\) extracted from the VEHICLE\_STATUS\_RESPONSE.

**DRIVER\_STATUS\_REQUEST** = \([x_1, d_{lid}, v_{lc\text{ategory}}]\)

DaV-DB creates a DRIVER\_STATUS\_RESPONSE containing information such as the sequence number \(x_1\) of the request, the identity of the requested driver \(d_{lid}\) and the driving license status \(d_{lstatus}\) of the driver. If the driver has any conditions \(c_1, \ldots, c_x\) regarding his or her driving license, these conditions are stored in DRIVER\_CONDITIONS. DaV-DB returns the DRIVER\_STATUS\_RESPONSE to DaV-VS.

**DRIVER\_STATUS\_RESPONSE** = \([x_1, d_{lid}, d_{lstatus}, \text{DRIVER\_CONDITIONS}]\)

**DRIVER\_CONDITIONS** = \([c_1, c_2, \ldots, c_x]_{x=0, \infty}\)

**Driver status**

<table>
<thead>
<tr>
<th>Status</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>0</td>
</tr>
<tr>
<td>No driving license</td>
<td>1</td>
</tr>
<tr>
<td>No driving license for this vehicle category</td>
<td>2</td>
</tr>
<tr>
<td>License temporarily revoked</td>
<td>3</td>
</tr>
<tr>
<td>Not renewed</td>
<td>4</td>
</tr>
<tr>
<td>Person not found</td>
<td>5</td>
</tr>
<tr>
<td>Error</td>
<td>6</td>
</tr>
</tbody>
</table>

Now DaV-VS has received the required information and creates a DRIVE\_RESPONSE message.

**DRIVE\_RESPONSE** = \([x_1, v_{status}, d_{lstatus}, \text{DRIVER\_CONDITIONS}, \text{ResponseCode}]\)

**DRIVER\_CONDITIONS** = \([c_1, c_2, \ldots, c_x]_{x=0, \infty}\)

The ResponseCode could be a list of possible results:

0: Reject start-up
1: Accept start-up
2: Outdated request
3: etc...

DaV-VS encrypts the response message with $v_1$’s public key $v_{1public}$ so that only $v_1$ can read the drive response.

\[
\text{ENCRYPTED\_DRIVE\_RESPONSE} = \text{encrypted} (\text{DRIVE\_RESPONSE}, v_{1public}).
\]

DaV-VS signs the encrypted drive response message with its private key $DaV\text{-VS}_\text{private}$ to ensure that the receiving vehicle can authenticate the sender of the encrypted drive response.

\[
\text{SIGNED\_DRIVE\_RESPONSE} = \text{signed} (\text{ENCRYPTED\_DRIVE\_RESPONSE}, \text{DAVVS}_\text{private})
\]

![Figure 40 - SIGNED\_DRIVE\_RESPONSE](image)

The signed and encrypted drive response message is sent to $v_1$.

**Verifying and extracting information from a DRIVE\_RESPONSE**

$v_1$ receives the signed and encrypted drive response and verifies the signature using $DaV\text{-VS}_\text{public}$.

Thereafter, $v_1$ decrypts the encrypted response message using $v_{1private}$, thus extracting the DRIVE\_RESPONSE.

\[
\text{Decrypt} (\text{ENCRYPTED\_DRIVE\_RESPONSE}, v_{1private}) \Rightarrow \text{DRIVE\_RESPONSE}.
\]

![Figure 41 - DRIVE\_RESPONSE](image)

$v_1$ can now be certain that the DRIVE\_RESPONSE actually was sent by the trusted $DaV\text{-VS}$ and that it was intended for $v_1$. As a last precaution, to separate multiple requests from the same vehicle, the random sequence number $x_1$ in the DRIVE\_RESPONSE is compared to the DRIVE\_REQUEST to determine which response has arrived.

The vehicle checks $v_{1status}$ to ensure that the vehicle is allowed to be driven. The vehicle also checks $d_{1status}$ to ensure that the driver is allowed to drive the vehicle. Furthermore, the vehicle must check whether there are any driver conditions in DRIVER\_CONDITIONS that have to be met for the driver to start the vehicle.

Response received... analysing results.
Vehicle status... OK!
Driving license status... OK!
STARTING VEHICLE!
7.3.2 Nonpermanent network connection

A real-life implementation of an EDL framework has to be able to handle the fact that network access is, or could be, nonpermanent. As with almost every other existing networking technology, factors such as software or hardware errors, lack of coverage, power outage and physical malfunction can result in a vehicle being unable to communicate with remote databases or other remote entities. These network interruptions must be handled within an EDL framework to ensure that the implementation of the framework doesn’t result in total meltdown if network access is interrupted.

In the previously described scenario, in the online license verification with permanent vehicle networking approach, every vehicle had to send a request and wait for a response from a remote verification database to start the vehicle. In this approach, vehicles simply would not start if they lacked remote access to the driver and/or vehicle verification service.

With an online license verification approach handling nonpermanent vehicle network access, the protocol should utilize acknowledgement messages and timeouts to handle malfunctioning or nonexistent remote access.

Such implementation could look like the following:

![Diagram of nonpermanent network EDL verification](Figure 42 - Nonpermanent network EDL verification)

Unable to communicate with DaV-VS, no acknowledgement:
These timeouts and acknowledgements handle interrupted network access but do not provide any way of verifying licenses whenever network access is interrupted. The options are to

- allow the driver to start the vehicle without proper online license verification,
- prohibit the driver from starting the vehicle because online license verification hasn’t been performed or
- set up an environment to be able to perform offline license verification (see section 7.4).

To allow start-up of the vehicle without license verification, has a major drawback: a driver without proper driving license is able to start the vehicle. An approach including the first and second options would result in either vehicles not starting, even if a driver has a valid driving license, or drivers without proper licenses for the vehicle category being able to drive vehicles if the network connection is somehow not working properly. This approach could result in a security issue where drivers without proper licenses manipulated vehicle network access so that the vehicle could be started without a proper license.

An offline license verification option is presented in section 7.4.

The first and last options, to allow a driver to start the vehicle without any license verification and to perform offline verification, could be combined in situations when the network connection not working properly. Such situations can be handled by requiring the driver to digitally sign a drive request or a digital signature of the preferred action. Such signatures of preferred actions would be methods to keep track of actions performed, such as starting and driving a vehicle, without performing complete online license verification. The driver can be informed of the nonexistent remote access and that only offline verification is performed. The signed drive request can be used to prove that the driver was aware of the offline verification and that he or she was responsible of verifying the driving license.
7.4 Offline EDL verification

The offline verification methodology, as opposed to the online verification methodology, does not require any vehicle to connect to an online driving license database. This approach reads the license information from the driver’s smart card and then verifies this information with the vehicle’s built-in requirements regarding driving license category.

This methodology has some potential points of failure regarding verification. One situation would be when a driver holds a driving license that has been revoked at some point. Let us say that the license was revoked but the driver claims to have lost the physical smart card, so the contents of the smart card cannot be modified by any governing authority. The government can see in the databases that the license has been revoked, but when a driver uses an EDL in an offline verification application, up-to-date driver license information will not be available. Therefore, there is a possibility, a risk, that drivers could start a vehicle without the proper driving license category.

At some point the vehicle’s EDL reading device has to be fitted with the digital certificate of the trusted CA or CAs that issue the driving licenses. This requires that every license-issuing CA be included.

This offline insertion of digital certificates has the drawback of not being able to determine whether a digital certificate has been revoked or a driving license suspended. A total offline verification methodology thus requires some kind of plug-in synchronization at regular intervals to be able to guarantee long-term accuracy. These software updates could possibly be done at an annual vehicle inspection or vehicle service.

This verification methodology requires that the police and other driving license–revoking authorities manually update the information on the smart card or deny access to the card.

In contrast to the online license verification approach, where license information is stored remotely, separate from the vehicle’s own data, an offline license verification approach would mean that driving license information is stored locally. This driving license information could be stored either in the vehicle or in the EDL presented by the driver. A combination of both these storage methods could be used.
The on-card offline license verification method is perhaps the most simplistic way of verifying driver license information regarding the amount of data that has to be stored. The driving license information is provided by each driver who presents his or her EDL to the vehicle, and the vehicle has to extract only the driving license information from the EDL.

The on-board offline license verification method means that the driving license information for every driver that can drive the vehicle must be stored within the vehicle’s own data storage. The complexity of such data is increased when more driving license information is stored within each vehicle. If a vehicle should be able to be driven by everyone that has proper driving license for that specific vehicle category, all those driver records have to be included in the on-board license verification information. This locally stored license information could also lead to privacy issues, which are further discussed in the privacy chapter.

The on-board license verification information could be stored in a local database containing both driving license information and vehicle information.

### 7.4.1 On-card offline license verification

The methodology is based on the driver being issued an EDL that represents his or her issued driving license with the driving license information stored in the smart card.

This driving license information can be stored within either the public or private data storage area within the EDL. If the driving license is not password protected and control of this information is the only verification done, then the EDL verification would work like any other key operating the vehicle. This would introduce minimal security into the EDL framework. Some kind of knowledge factor would be required to guarantee validity. The system would require at least a password. The password would, hopefully, be known only to the holder of the license, increasing the level of security beyond key-only security level. Every driver should be exclusively responsible for keeping his or her password a secret.

The offline start-up sequence would be the following:

Driver $d_1$ requests to drive vehicle $v_1$.

$v_1$ has VIN $v_{1\text{vin}}$, license plate number $v_{1\text{plate}}$ and vehicle category $v_{1\text{cat}}$.

$d_1$ has a valid driving license for $v_{1\text{car}}$.

$d_1$ possesses an EDL $EDL_{dl}$ for $v_{1\text{car}}$.

$d_1$ pushes the start button (or turns the ignition key).

$v_1$ detects that an unknown driver $d_1$ requests to start and drive $v_1$.

$v_1$ prompts the unidentified $d_1$ to connect an EDL to the vehicle’s EDL reader.

Please insert your EDL into the reader or move the contactless EDL within range of the reader if you would like to drive vehicle ABC123 (VIN: 123456789) of vehicle category XX.

$d_1$ connects $EDL_{dl}$ to the EDL reader of $v_1$.

Now the offline driving license verification approach can utilize two verification methods. The first one is to let the user digitally sign a DRIVE_REQUEST containing relevant information regarding the driver’s wishes to drive the vehicle.

This SIGNED_DRIVE_REQUEST is then inspected by the vehicle, which verifies the digital signature. This requires that the vehicle gain access to the driver’s public key and that the vehicle trust the CA that has issued the certificate including the driver’s public key. This could be resolved by including the digital certificate within the signed DRIVE_REQUEST returned to the vehicle. When the vehicle
verifies the digital signature of the driver, the vehicle checks the certificate’s extension fields of the driver’s digital certificate to see whether the driver has a valid driving license for the current vehicle.

![Diagram of the process of PKI digital signature of a DRIVE_REQUEST]

The first steps of the offline license verification approach are identical to the steps within the purely online license verification framework. $v_1$ creates a DRIVE_REQUEST containing a timestamp $t_1$, VEHICLE_ID and a random sequence number $x_1$. The VEHICLE_ID contains the vehicle’s license plate number $v_1$plate_presented and the vehicle’s VIN $v_1$vin_presented.

\[
\text{DRIVE_REQUEST} = [t_1, \text{VEHICLE_ID}, x_1]
\]
\[
\text{VEHICLE_ID} = [v_1\text{plate}_{\text{presented}}, v_1\text{vin}_{\text{presented}}]
\]

The DRIVE_REQUEST is presented to the driver, and he or she is prompted to sign it. The vehicle prompts $d_1$ to connect the EDL to the EDL reader so that a digital signature can be created. Since the private key of the driver demands CHV1, $d_1$ has to provide the EDL reader with a PIN, biometric data or other unlocking and authentication method to generate a signature.

When the user is authenticated and the private key accessed, the EDL reader signs the DRIVE_REQUEST, thus creating a SIGNED_DRIVE_REQUEST. The SIGNED_DRIVE_REQUEST also contains the driver’s digital certificate, which will be essential when verifying that the driver actually holds a valid driving license for the current vehicle.

The EDL reader of $v_1$ prompts $d_1$ to identify himself or herself by password, PIN or biometrics to be able to access (CHV1) the private key of $d_1$ ($d_{\text{private}}$) securely stored within the EDL.

Enter PIN or press finger against scanning device.

\[
\text{SIGNED\_DRIVE\_REQUEST} = \text{signed(DRIVE\_REQUEST, } d_{\text{private}})\]

The SIGNED_DRIVE_REQUEST is then returned from the EDL reader (signing is performed within the smart card) to the vehicle. The vehicle verifies the signature of the driver. For the vehicle to verify the signature of the driver, it has to trust the issuer (CA) of the driver’s public certificate. When the
vehicle has determined that the signature of the DRIVE_REQUEST is valid, the vehicle can move on to examining the extensions of the driver’s public certificate. The vehicle checks to see whether the certificate extension holding the license information allows the driver to drive the vehicle.

If $v_{\text{cert}} = d_{\text{cert_ext.lic}}$ then start the vehicle.

The second on-card offline verification approach utilizes the storage of a complete EDL on the smart card. This EDL could be stored in plain text, XML or ASN.1. For security reasons some parts of the EDL should be read-only to the user, such as driving license information regarding which vehicles the driver is allowed to drive. If the driver had read and write permissions regarding such information, this would allow for illegal modification of a driver’s license. This verification approach identifies the driver through some authentication procedure, such as PIN or biometrics.

Every driver verification method should be combined with proof-of-possession of a physical EDL smart card and the knowledge of a PIN or biometrics of the holder of the EDL. If this authentication were not used, it would be possible to find, steal or copy someone’s EDL.

A choice of on-card offline methods will be presented as a solution to the following online/offline adaptable EDL system. The following sections will show an EDL system that utilizes both stored EDL data and the PKI features of the smart cards. Note that CHV1 or CHV2 could be done in any of the abovementioned driver authentication methods.

### 7.4.2 On-board offline license verification

To ensure that network interruptions do not affect the function of an EDL framework, some portion of the updated license and vehicle information should be stored locally within every vehicle used through the framework. The challenge regarding this approach is to maintain up-to-date information in every vehicle and to determine which information should be stored.

Ideally, every included vehicle would maintain an updated local copy of a complete license and vehicle database at all times. As a result of network interruptions and for possible storage and privacy reasons, this would not be realistic or practical. The on-board information database should include the following:

- Updated license information regarding drivers that are allowed to drive the vehicle
- Updated vehicle information regarding the vehicle

Determining the number of drivers included in the local license information storage could be difficult. If the driver lends a vehicle to a person not included in the database, the system would not allow this driver to start the vehicle.

Determining the required information regarding the vehicle is a much easier task as it is a specified entity storing its own limited information.

A local database has to be updated as often as possible to be an “updated” local driver and vehicle information database. A pure offline approach regarding on-board data storage would require vehicles to update the local databases at regular intervals. Maintaining local databases require vehicles to retrieve updated information from a reliable data source. Therefore, an online/offline adaptable EDL verification approach combining technology from both the online and offline methods previously described would be most suitable in an EDL framework implementation.
7.5 Online/offline adaptable EDL verification

A conclusion regarding online versus offline verification reveals weaknesses in both verification methodologies, thus implying that maybe a combination of them would be the best possible approach.

When a network connection is available, the online and offline verification are performed. If the vehicle lacks a network connection, only the offline verification is performed. The offline-only approach should then also include local vehicle and driver license information storage to achieve as high reliability as possible.

This local information storage could be a subset of the complete remote database containing all driver and vehicle information. Such local information could be updated at regular intervals depending on the availability of the network connection or similar connectivity.

A combination of all previously presented approaches would be a framework that verifies license and vehicle information online using a verification service whenever the network is in operation. The license and vehicle verification is done on-board locally whenever remote access is not available. This local information is updated whenever remote networking is in use. This remote networking could be wireless or physically connected. Manual updates could also be performed using plug-in information.

The complete EDL framework presented within this report will utilize the verification methods from both the online and offline approaches. The EDL system adapts to the current conditions and makes real-world implementation possible.

The complete EDL system would do the following:

- Always check driving license data stored on the smart card connected to the EDL system.
- In online mode, use the online verification system.
- In offline mode, use local database verification if the database is updated.

**DRIVE REQUEST**

Driver \( d_1 \) requests to drive vehicle \( v_1 \).

- \( v_1 \) has VIN \( v_{1\text{vin}} \), license plate number \( v_{1\text{plate}} \) and vehicle category \( v_{1\text{cat}} \).
- \( d_1 \) has a valid driving license for \( v_{1\text{cat}} \).
- \( d_1 \) possesses an EDL \( EDL_{d1} \) for \( v_{1\text{cat}} \).
- \( d_1 \) pushes the start button (or turns the ignition key).

\( v_1 \) detects that an unknown driver \( d_1 \) requests to start and drive \( v_1 \). The EDL system detects that no valid driver registration has yet been registered for this starting attempt, and therefore the EDL system computer will not pass on a “start command” on the CAN bus.

\( v_1 \) requests that \( d_1 \) insert his or her EDL into the smart card reading device (or that the smart card be put within reading range if it is a contactless or dual-interface smartcard).

Please insert your EDL into the reader or within the range of the reader if you would like to drive vehicle ABC123 (VIN: 123456789) of vehicle category XX.

When the smart card EDL is connected, the start-up sequence can begin.

The EDL system reads the license information of the driver’s smart card. If the smart card does not contain proper driving license information, the start-up sequence is aborted.
The driving license information is read from \( \text{EF}_{\text{EDL LIC}} \) (or \( \text{EF}_{\text{EDL}} \) depending on the storage design; the presented sequence assumes separate XML data file storage; see section 12.1).

*For security reasons the system would require CHV2 from the user if the contactless interface is used, to prevent unwanted access to an EDL within range. The connected interface does not require the user to present valid CHV2 if the access control is set to ALW.*

Improper driving license information (\( \text{EF}_{\text{EDL LIC}} \) contains \text{holder} element data \text{false} within the current vehicle section; see XML representation) would result in the vehicle not starting.

Driver \text{FIRSTNAME} \text{LASTNAME} (YYMMDD-XXXX) has not presented a valid driving license for vehicle category XX of ABC123 (VIN:123456789).

If the driving license information is valid (\( \text{EF}_{\text{EDL LIC}} \) contains \text{holder} element data \text{true} within the current vehicle section), the driver has a driving license that gives him or her right to drive a vehicle of the current vehicle category. Hence, the start-up sequence will continue.

Welcome, \text{FIRSTNAME} \text{LASTNAME} (YYMMDD-XXXX).

A valid EDL regarding vehicle category XX has been presented.

Start-up sequence initiated...

\( v_f \) scans the network connection to determine whether the vehicle is in *online* or *offline* mode.

If the vehicle is in offline mode, \( v_f \) scans the local database to see if it is *updated* or *outdated*.

This results in four possible scenarios:

<table>
<thead>
<tr>
<th>Networking</th>
<th>Local DB</th>
<th>Updated</th>
<th>Outdated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online</td>
<td></td>
<td>Section 7.5.1</td>
<td>Section 7.5.2</td>
</tr>
<tr>
<td>Offline</td>
<td></td>
<td>Section 7.5.3</td>
<td>Section 7.5.4</td>
</tr>
</tbody>
</table>

### 7.5.1 Online access and updated local database

The driving license verification is done online, and the local database is used only to verify that the user is allowed by the vehicle’s administrator (often owner) to drive the current vehicle.

If network connection is lost during the online verification, the verification is done according to section 7.5.3, “Offline access and updated local database”.

Online access verification is performed as in “Permanent network connection” (section 7.3.1). Note that the EDL reader will require cardholder verification 1 (CHV1) to be able to create a \text{SIGNED\_DRIVE\_REQUEST}.

When the vehicle has decrypted and read the returned \text{DRIVE\_RESPONSE}, no more license information is required from the local database.

The \text{DRIVE\_RESPONSE} data is read, and the vehicle will start if the driver has the correct driving license.
7.5.2 Online access and outdated local database

This verifies the EDL as in “Online access and updated database” (section 7.5.1).

If network connection is lost during online verification, the verification is done according to section 7.5.4, “Offline access and outdated local database”.

Another difference is that the EDL system, after completion of the verification sequence, will request updated database data from the EDL verification service.

7.5.3 Offline access and updated local database

As the vehicle does not have a working networking connection, it will not be able to communicate with an online DaV-VS. The EDL system has determined that the locally stored driver database is up to date and can be used for verifying the driver’s credentials.

**Figure 47 - Local database EDL verification**

Vehicle $v_j$ extracts the personalIdentificationNumber from the $EF_{DL, Lic}$ and inserts the data into a query that is sent to the local database. The database responds with a driver license response containing the issued license information $d_{i\text{status}}$ regarding driver $d_i$.

$v_j$ also extracts the vehicle’s VIN and sends that data to the local database. A vehicle response message is returned indicating the status $v_{i\text{status}}$ of the vehicle.

$v_j$ extracts the time of the start-up request performed by the driver, $time_{dl,v_j}$.

$v_j$ creates a DRIVE_REQUEST containing the personalIdentificationNumber, the $VIN_{i_j}$ and the $time_{dl,v_j}$.

$$\text{DRIVE\_REQUEST} = [\text{personalIdentificationNumber}, \text{VIN}_{i_j}, \text{time}_{dl,v_j}]$$
The EDL system presents the DRIVE_REQUEST information to the driver and prompts the driver to digitally sign the DRIVE_REQUEST. To be able to use the private key $EF_{pk}$ (driver’s private key) stored on the smart card, the user will have to undergo cardholder verification 1 (CHV1).

Would you, FIRSTNAME LASTNAME (YYYYMMDD-XXXX), like to drive vehicle ABC123 (VIN: 123456789) at local time YYYY-MM-DD HH:MM:SS? Please verify by digital signature!

Signing the DRIVE_REQUEST creates a SIGNED_DRIVE_REQUEST.

$$\text{SIGNED\_DRIVE\_REQUEST} = \text{signed(DRIVE\_REQUEST, } d_{\text{private}}\text{)}$$

The SIGNED_DRIVE_REQUEST is stored within the local database for future use.

If the driver and vehicle response contains a positive match regarding the driver’s issued credentials and a verification that the vehicle status is confirmed, the vehicle will start.

### 7.5.4 Offline access and outdated local database

If the vehicle is in offline mode and the database is outdated, the EDL system could be set to respond in one of these ways when implementing the EDL system:

- Reject start-up of the vehicle.
- Accept start-up of the vehicle if the EDL information on the smart card is correct and the driver digitally signs a “knowledge verification”. This knowledge verification is stored within the vehicle’s EDL system and is later used by the legal system to prove that the driver knew that access to the online license verification service was not available and that the local database was outdated. The driver, as part of signing the knowledge verification, was informed that he or she was responsible for checking whether his or her driving license was up to date.

The first choice would be the safest in terms of avoiding misuse and thus preventing unlicensed driving.

The alternative, to permit start-up of the vehicle after creating a signed knowledge verification, could possibly be misused, but the misuse would be clearly documented, and this knowledge verification could be used to prove the intent of an unlicensed driver.

Vehicle $v_1$, after extracting the license information from the connected EDL’s file $EF_{EDL\_lic}$, continues to present the “knowledge verification” to the user.

No online access available...
Local database outdated...

Connected EDL is valid. Please sign a knowledge verification and be aware that you are responsible for verifying that you, FIRSTNAME LASTNAME (YMMDD-XXXX), have the correct driving license to drive vehicle ABC123 (VIN: 123456789) category XX at time YYYY-DD-MM HH:MM:SS.

The KNOWLEDGE_VERIFICATION contains the personalIdentificationNumber contained in the EDL and the vehicle information, and is signed by the driver’s private key $d_{\text{private}}$.

$$\text{KNOWLEDGE\_VERIFICATION} = [\text{personalIdentificationNumber, VIN}_1, \text{time}]$$

$$\text{SIGNED\_KNOWLEDGE\_VERIFICATION} = \text{signed(KNOWLEDGE\_VERIFICATION, } d_{\text{private}}\text{)}$$
If the driver signs the KNOWLEDGE_VERIFICATION and if it is successfully stored, the vehicle will start.

### 7.5.5 Change in online/offline access

If a change in online/offline network access is detected after a start-up sequence has been initiated, the following will be the case:

**Network ONLINE to OFFLINE**

The connection is lost, and the EDL system restarts the verification sequence at the corresponding offline sequence (continues after the connection of the EDL). For example, if the online access and updated local database verification approach is initiated and the connection is lost, the EDL system restarts the verification steps and utilizes the verification defined in the offline access and updated local database.

**Network OFFLINE to ONLINE**

No change in verification approach. The system could send the SIGNED_DRIVE_REQUEST to the DaV-VS at a later time to do a full online verification. Offline to online access will not restart the start-up sequence, thus guaranteeing that fluctuating access would not cause the start-up sequence to continuously start over.

**Database UPDATED to OUTDATED**

No change in verification approach. When the EDL system checks whether the online driver and vehicle database is updated, this result would be valid until the start-up sequence is completed.

### 7.6 Update of local database

The local database should be updated at some interval if network access is provided. When a database should be declared as outdated and has to be updated is a design issue affected by the level of security needed versus the amount of information sent and received.

A possible design would be to update the local database at least at the annual inspection of the vehicle. This would be a minimal requirement, and additional updates should be mandatory.

Offline vehicles and vehicles driving without updated local databases could show signs of these conditions, for example by:

- visual notifications (interior or exterior warning lights) or
- audio notifications (interior or exterior alarms or signals)

### 7.7 Negative validations

Some possible scenarios that would result in negative validations are as follows:

![Figure 48 - DRIVE_RESPONSE](image-url)
DRIVE REQUEST PRESENTED LICENSE PLATE OR VIN NOT CORRECT

If \( v_{\text{plate}_{\text{presented}}} \neq v_{\text{plate}} \) or \( v_{\text{vin}_{\text{presented}}} \neq v_{\text{vin}} \), create a RETURN_MESSAGE containing a response code 
\( c_{\text{error}_{\text{presented}_{\text{plate}}} \text{ or } c_{\text{error}_{\text{presented}_{\text{vin}}} \text{ and previously received random number } x_1. \text{ This would indicate that the vehicle has not presented the correct VIN or license plate number to the user for signature.}} \)

Response code = incorrectVIN or incorrectPlate.

DRIVE REQUEST TIME INVALID

If \( t_l \) is not within the allowed time span, create a RETURN_MESSAGE containing a response code 
\( c_{\text{time}_{\text{due}}} \) and previously received random number \( x_1 \).

Response code = driveRequestUnvalidTime

DRIVER REJECTED

\( d_l \) is rejected; see \( d_{\text{status}} \) for explanation.

VEHICLE REJECTED

\( v_l \) is rejected; see \( v_{\text{status}} \) for explanation.

DRIVER AND VEHICLE REJECTED

Both \( d_l \) and \( v_l \) are rejected; see \( d_{\text{status}} \) and \( v_{\text{status}} \) for explanation.
Chapter 8 EDL framework usage

This chapter presents some key aspects of the possibilities of usage when deploying an EDL framework in today’s traffic environment. These more specific areas of usage have to be analysed in combination with the previously presented possibilities for reducing unlicensed driving throughout the world.

8.1 Information advantages

The EDL framework (the smart card EDL and the remote driver and vehicle verification service) should provide a gateway for public information access after electronic identification. Information from the smart card EDL could be extracted, as well as information from the DaV-VS or DaV-DB after proper PKI identification is done.

An example: A person turns 16 years of age and has reached the legal age to begin driving practice vehicle category B (car). He or she logs on to the government’s database handling driving licenses and learner’s permits using his electronic identification card. He or she sees that he or she has not got a valid driving license for vehicle category B and is not permitted to drive such a vehicle alone. He or she also sees that he or she has not been granted permission to practice driving vehicle category B. The information should be clearly presented to the user, thus providing no ambiguity regarding interpretation of the provided data.

![Figure 49 - Example of clearly presented information](image)

Providing users of the EDL framework with crystal-clear information must be one of the most important aspects of the implementation. No one should ever be able to blame the EDL framework for any misconceptions or blame the system for any illegal activities.

8.2 Reduce drunk driving

The EDL framework could prevent drivers from starting vehicles without installed alcohol screening devices. Such drivers could have special conditions issued on their driving license, perhaps after being sentenced for driving under the influence of alcohol, allowing these drivers to drive only vehicles with alcohol screening devices. If the vehicle or the online verification service recognizes that the driver has such a limitation and that the vehicle lacks the installed alcohol screening device, the start-up of the
vehicle would be rejected. The start-up attempt would not be authorized unless the driver provided the alcohol screening device with a valid breath sample proving that he or she was not intoxicated beyond the legal limit.

8.3 Acknowledgement of misuse

If the EDL framework implements a smart card EDL approach, where every smart card EDL stores a personal PKI private key, then the framework could be used by the legal system to prove intent. The system could inform the user of the current status of the vehicle, traffic rules that affect the vehicle and its driver and other aspects that should be confirmed by a driver before being allowed to drive the vehicle.

Below are some examples:

The periodical vehicle inspection has not been performed, and therefore the vehicle has been given a “driving ban”\(^\text{20}\). Swedish legislation would allow the driver to drive the vehicle to a vehicle inspection centre or to an authorized auto repair garage. The vehicle could be started, even if there was a driving ban, if the driver digitally signed an acknowledgement that he or she understood that he or she was allowed to drive the vehicle only to such facility.

To start the vehicle:
Please digitally sign a verification that you, FIRSTNAME LASTNAME YYYYMMDD-XXXX, understand that the vehicle ABC123 (VIN:1123456789) has a driving ban (due to no periodical vehicle inspection) and you are allowed to drive the vehicle only to a vehicle inspection centre or an authorized repair garage.

Another example could be when a country’s legislation demands the use of mud and snow tires during some specified time of the year. When the vehicle is started for the first time within this period, the system would inform the driver of the legal aspects:

To start the vehicle:
Beware that vehicles of the current vehicle category (XX) are required to be equipped with mud and snow tires from YYYY-MM-DD to YYYY-MM-DD. Please sign acknowledgement.

The notifications could include almost every aspect of information that has to be proven in a court of law. The implementation of signed acknowledgements would greatly reduce the number of traffic violations resulting from drivers being unaware of the current situation.

---

\(^\text{20}\) “Körforbud”
8.4 Drive log

When using an EDL framework, the technology would provide a way to record most of the vehicle’s data in combination with driver data. Individual records could be stored regarding vehicle speed, coordinates and which driver was driving a vehicle at any time. This can be compared to the black box in airplanes. This drive log could be used as a crash recorder when a vehicle is involved in an accident. The police could retrieve data from the drive log to investigate accident scenarios.

Such drive logs risk violating people’s privacy, and this aspect is further discussed in the privacy chapter (see section 9).

Perhaps an EF on the EDL smart card could be used to store the last vehicle driven, when the vehicle was started and when the driver completed the drive:

$$\text{EF}_{\text{last\_drive}} = [v_{\text{vin}}, v_{\text{plate}}, \text{time}_{\text{start}}, \text{time}_{\text{stop}}]$$

This EF storage would require that the EF could be updated only by the EDL system and not by the holder of the EDL.

Previous chapters have also pointed out the possibilities to store DRIVE_REQUESTs or SIGNED_DRIVE_REQUESTs both in online verification system storage and in local databases. The SIGNED_DRIVE_REQUESTs provide a drive log entry truly proving that the driver signing the drive request actually was the one that intended to start the vehicle.

8.5 Advantages in police work

Everyday police work in traffic environments could greatly benefit from the implementation of an EDL framework. If we set privacy aside for a moment and focus mainly on crime prevention, the following could be said.

Police officers could know in advance who was driving a specific vehicle by inspecting the vehicle’s license plates and sending a request to the online driver and vehicle database. The database would inform the police officers about who signed the latest DRIVE_REQUEST and is therefore driving the vehicle.

Both online and offline verification approaches could be extended to include records of coordinates collected by GPS equipment. Controller databases could receive signals when drivers perform illegal activities such as speeding, driving in forbidden areas or against traffic, or not stopping at red lights. Previous tests, included within Dr. Goldberg’s research, have shown that remote stopping of vehicles is possible could most likely be used to reduce the number of dangerous car chases. It is of great importance to stop these dangerous car chases as they risk numerous of lives, including the driver, innocent bystanders and other people in traffic, as well as the police officers involved in the chase. If the police officers got information regarding who was driving the vehicle, they could also abort the pursuit and arrest and/or question the driver at a later time.

A great administrative advantage would be roadside access to driver and vehicle information. An EDL framework would simplify the issuing of tickets. A digital issuing device scanning the driver’s EDL would input all necessary information to the ticket-issuing system. As the police officer’s EDL also would be connected to the device, both the driver receiving the ticket and the police officer could digitally sign and countersign the electronic ticket. All information needed could be stored within the EDL.
8.5.1 Electronic issuing of tickets

The EDL framework could be used to implement an electronic ticketing system used roadside by law enforcement personnel. Tickets could be issued for speeding drivers, driving without a license or other traffic violations depending on the legislation of the country. Such roadside application could be used to issue tickets and to write reports that later would be sent to a prosecutor or other law enforcement personnel for further investigation.

![Figure 51 - Example of EDL information usage when issuing Swedish tickets](image)

The ticket should be signed by both the issuing police officer and the driver receiving the ticket. This validation guarantees nonrepudiation.

8.6 Driving with a learner’s permit

An EDL framework has to be able to handle what is probably the most common situation of unlicensed driving: unlicensed drivers driving with a learner’s permit together with a tutor. When it is done according to the country’s laws and regulations, this unlicensed driving is not illegal and must therefore be accepted.

The EDL system could be implemented so that it recognizes that a potential driver does not have the proper driving license to drive the vehicle alone and that the potential driver has only a learner’s permit on the smart card EDL and in the remote license database.

The EDL system could require that the DRIVE_REQUEST be signed by both the learner and the tutor.

The system could inform both the learner and the tutor of any restrictions that apply to the educational driving they are about to perform.

A simplified example could look like this:

No valid license presented...

Driver FIRSTNAME LASTNAME (YYYYMMDD-XXXX) has a valid learner’s permit for vehicle category XX for vehicle ABC123 (VIN: 123456789). Would you like to drive with the learner’s permit? Please digitally sign a drive request!

Drive request signed, thank you...

Please connect an EDL of a co-driver approved to supervise FIRSTNAME LASTNAME (YYYYMMDD-XXXX)...
Co-driver FIRSTNAME LASTNAME (YYYYMMDD-XXXX) is approved to supervise...
Co-driver, please digitally countersign drive request...
Drive request signed and countersigned, vehicle starting...

8.7 Driving with a lost valid driving license

As a result of ordinary human behaviour, sometimes things become lost or broken, such as driving licenses. In some of these cases, licensed drivers should be able to drive a vehicle even without being able to present a valid EDL.

For example, a driver might drive an EDL-connected vehicle to a remote location and then lose or break his or her EDL at that remote location. The EDL framework must allow licensed drivers to drive their vehicles for some restricted time or within some restricted area or some restricted number of times if they have lost their EDL or if it has malfunctioned.

One possibility would be to allow previously accepted drivers to start the vehicle without presenting a valid driving license for some restricted number of times. Such invalidated starts could be registered, and the limiting counter would have to be reset by presenting a valid EDL. If the EDL system receives any indication that an EDL was revoked, such “licenseless” start-ups should not be possible.

An online environment could even allow a driver to be identified by fingerprint and later or right away verify that he or she indeed has a valid driving license.

Licensed drivers should be able to drive vehicles during the time it takes for an issuer to issue a new EDL to the driver. A lost or malfunctioning EDL should not prevent the driver from driving vehicle categories for which he or she has a valid driving license. This should be handled by providing license holders with quick issuing services every day of the year, 24 hours a day, almost everywhere within the EDL zone, or the EDL framework should provide a means of trusting drivers and giving them a chance to present a valid EDL at a later time.

It is clear that handling exceptions in an accepted way is perhaps even more complicated than handling the ordinary validation of EDLs. To implement a system that simply turns on a red or green light based on whether the driver presents a valid driving license is simply not enough.

8.8 Driving in emergency situations

Emergency situations arise constantly in many different ways. Some situations are life threatening, and as a result, rules and regulations must sometime be set aside in order to save human lives. What would happen if an EDL-licensed driver could not present a valid driving license in a life-threatening situation, for example, after suffering from a heart attack? In many life-threatening or other emergency situations, it is vital to be able to drive a vehicle from point A to point B, possibly to a hospital or other facility, in order to save human lives.

The EDL framework should present users with the ability to start a vehicle without any form of proof-of-possession regarding a valid driving license in order to handle such emergency situations. These “emergency start-ups” should be handled with care to prevent misuse of this alternative start-up procedure. One simple way to handle such emergency start-up would be to enable the vehicle’s hazard lights and/or periodically sound the vehicle’s horn or flash the headlights. The possibilities are many, but the surrounding traffic should be warned that a potentially unlicensed driver is driving. Such emergency start-up should also be logged within the vehicle and sent to an online verification database or register if an online approach is implemented. These offline and possibly online records, in combination with the visual and audio warnings, should improve the controlling authorities’ chances of preventing misuse.
Chapter 9 Privacy and information security

This chapter discusses and analyses the privacy and information security issues regarding the presented EDL framework and its included parts. This chapter focuses mainly on the risks of intentional misuse by individuals, organizations and authorities.

9.1 PKI risks

The presented EDL framework is in many aspects based on the security of PKI asymmetric encryption. The strength of asymmetric encryption depends on the key length chosen. If someone were able to resolve somebody else’s private key, the security of the system would be greatly reduced.

A private key can be compromised by different lines of attack. The key can be guessed by brute force, by evaluating either passwords or keys step-by-step depending on their length.

Another apparent security risk is the case when someone in or nearby a vehicle eavesdrops on the driver entering the PIN or password when accessing the private key stored on the EDL smart card.

The physical layout of most vehicles poses security threats regarding keeping driver’s PIN or passwords private. The position of a key panel affects the potential for key compromising. The risk of key compromising as a result of the non-private physical layout of a vehicle carrying multiple people would perhaps result in an EDL design consisting of biometrics CHV. Even if a passenger or another spectator could see the driver using a fingerprint reader or retina scanner, the observer would not be able to duplicate the information.

Another key issue in the security of the PKI-based EDL framework is the security of the trusted CAs, the management of stored CA certificates and the reliability of the CAs. It is crucial that only reliable trusted CA certificates be added to the EDL system. This goes for the client side, the vehicles, and the server side, the online verification certificate-storage system. Drivers should not be able to add user-defined CA certificates on their own. The root CA should be stored and modified only by system administrators.

Both vehicles and verification services must verify certificates against revocation lists or by OCSP to guarantee validity.

Intentional misuse, such as a driver starting a vehicle for another person, is more difficult to avoid. This misuse could technologically be reduced by implementing, for example, weight measurements of the person sitting in the driver seat or other continuously performed biometric scans where data is collected during driving of the vehicle. To guarantee a framework without any potential for misuse would require a system that collects a great amount of data at every moment by differentiated methods, probably reducing the driver’s privacy to a minimum.

9.2 Drive logs and other information

A major privacy issue is the logging of information such as drive requests, drive logs, driving patterns and other information linked to drivers and vehicles. People’s privacy can be invaded if this type of information is misused. The need to collect and store information has to be set in relationship to the respect of people’s privacy. These key aspects have to be considered:

- What type of information is stored?
- How is the information stored?
• Who has access to the information?
• How much of our privacy and private information are we willing to give up in order to improve traffic safety?

Designers of an EDL framework could justify mandatory information collection with the claim that driving a vehicle, as well as obtaining a driving license, is voluntary. Everyone has the right to avoid driving vehicles or not obtain a driving license. But people with opposing opinions could argue that driving a vehicle and obtaining a driving license are almost mandatory to be able to participate in much of everyday life in many places of the world. In some parts of the world, depending on the geography and other aspects of society, a life without a vehicle and/or driving license would not be possible. In such cases, people would not be able to choose to be excluded from participating in the logging of information within the EDL framework.

The range of information collected could vary from practically nothing to almost every piece of information available regarding the driver, his or her driving patterns and information regarding the vehicle. History has revealed many cases where information has been misused by entities such as governments and other authorities. The designer of the EDL framework has to be aware that extreme data collection approaches could provide authorities or other organizations with total knowledge of a person’s private life if they were given access to the acquired information.

Another aspect of trust is when a driver drives an EDL vehicle across multiple countries. A design issue is whether there should be one single worldwide verification service or different verification services across different geographic areas. Are all governments and organizations as trustworthy as the ones we trust within our own geographical region? Even if the driver resides in a country governed by a trustworthy government, what if the information sent to the roaming verification service were misused? And if a shared verification service were to be deployed, how would we ensure that every country included in the joint cooperation respected the stated regulations throughout the EDL framework?

### 9.3 Manipulating vehicles

The reliability and security of the complete EDL system is greatly dependent on the EDL reader and its connection to the vehicle’s electric system. Manipulating vehicle technology, including the EDL reader, would compromise the reliability of the EDL system. It is crucial that the vehicle present valid information to the driver, as he or she digitally signs the information presented. An EDL framework should use secure signature-creation devices (see section 3.2.3) connected to vehicles’ electric systems to minimise the risk of information manipulation.

Manipulated vehicles would potentially pose the risk of drivers being able to drive a vehicle without fulfilling the EDL framework’s driver requirements. Such nonvalidated driving could be done both by drivers unaware of their actions and by drivers that intentionally manipulate vehicles to do so.

The design of an EDL framework must prevent drivers and other people from manipulating vehicles equipped with EDL readers by, for example, bypassing EDL readers, preventing online verification by terminating network connectivity, or changing vehicle data, such as the vehicle category.

An implementation of an EDL framework would result in an increase of information handled by the included vehicles. This increase of information would most likely increase the risk that someone is willing to eavesdrop on the information passing through and stored within the vehicle’s computer. An increase of information handled by a vehicle’s computer could turn it into a potential target for hackers or other information-seeking entities. Vehicle computers might have to be fitted with antivirus applications and firewalls to protect them from digital attacks.

Even if the system designers were to digitally protect the vehicle computers, they would also have to guard against someone simply removing a storage device from the vehicle to extract information.
9.4 Biometrics and privacy

Biometric driver authentication could, depending on the information collected and how it is stored, represent a serious intrusion of privacy. How would people react if they had to give fingerprints, retina scans, DNA data, photographs and other personal data to be issued an EDL? The amount and type of collected data has to be weighed in relation to increased traffic safety.

There is a difference between leaving your fingerprint information with an authority that keeps records of all collected data and simply storing your fingerprint data on a smart card for personal use only. But biometric data stored in a private smart card could be accessed by external entities when a driver has to hand over his or her EDL, depending on who has access to read or write information on the card.

Biometric information is a wide spectrum of private data. Almost every aspect of a person’s private life potentially could be mapped if one were to extract all biometric data that is possible to collect. If such private data were extracted by malicious external persons, organizations or authorities, the potential for misuse would be massive.

There is an obvious difference between the public’s acceptance of the collection of different biometric data. Many people today use fingerprint readers for identification and as a way to access data and locations. But what is accepted by the general public? Is it acceptable to provide the authorities your fingerprint information or your DNA profile? Would people accept providing authorities with their retina scan data or weight information?

As described in previous sections, it is a matter of what type of information is stored, how the information is stored, who has access to the information and how much of our private lives we are willing to sacrifice to improve traffic safety.

9.5 Offline license verification issues

The offline verification approach included local information storage, or a database, that could contain driver license information and be stored within the vehicle. This local information has to contain license information regarding at least the people that should be able to drive the vehicle. A potential privacy and storage issue is when a vehicle may be driven by anyone. This approach could result in a large amount of information regarding many different people being spread across multiple vehicles.

It is important that private information cannot be exported and read, to ensure individuals’ privacy. Should such information be stored in encrypted form or only hash values of the original data?

The possibilities for information sharing and information storage could be greatly affected by the issuing countries’ legislation and privacy statements.

The EDL framework must also guarantee that private information stored on a smart card is kept confidential. Losing an EDL smart card or using an EDL in an unknown vehicle must not result in the information being extracted and viewed by external parties.

For example, imagine a country governed by an untrustworthy government or authority collecting all data from the EDL-connected vehicles sending and receiving data within its territory. Governments could also order validating officers to collect all offline data available when performing vehicle inspections or other roadside controls.

9.6 Law enforcement information

How would the general public react to the possibility that law enforcement officers could write information to their personal EDL? Would it be seen as a violation of privacy to store personal information in the data storage area of the EDL smart card?
Would the stored information be readable by the holder of the EDL, or should the information be read only by law enforcement officials? Could law enforcement information stored within the license holder’s EDL be of different types? Perhaps some information should be readable to the holder and some not. For example, could a police officer record that he or she has given the license holder a warning regarding prohibited vehicle equipment or reckless driving? Such registered warnings stored within the holder’s EDL should be viewable both by the license holder and by other law enforcement personnel. To make such warnings viewable, but not editable, by the license holder would increase awareness and possibly further increase traffic safety. On the other hand, perhaps some information should not be readable by the license holder, for example, notices regarding people involved in organized crime that travel across different geographical areas. Law enforcement agencies could store a hidden message in the person of interest’s EDL stating that the license holder is in fact a member of an organized crime ring and that the investigating personnel should be cautious.

This law enforcement information raises another question: can the law enforcement agency be trusted? History has revealed many examples all over the world when law enforcement personnel have proved to be untrustworthy or corrupt. Who inspects the actions of those who inspect the EDL information?

### 9.7 Medical information on EDL

What medical information, if any, would it be acceptable to include on the EDL smart card? What medical information would be suitable to register and share with various levels of data accessibility?

The lifesaving potential of medical information, as well as donor information, included on an EDL is possibly huge. But designers have to decide who should have access to the stored medical information. Perhaps a solution would be to let the EDL holder choose what medical information is stored on his or her EDL and moreover decide the level of data accessibility. For example, a holder might let anyone accessing the EDL see that he or she is epileptic so the person could perform lifesaving actions if the holder had a seizure, but might not let anyone know that he or she is not willing to donate organs.

At deployment of an EDL framework, the designers have to choose whether the registration of medical information should be mandatory or strictly on a voluntary basis. Or should medical information even be included in the EDL? It is possible but not mandatory for the functionality of the EDL framework.

### 9.8 What you see is what you sign

There is a possibility that a vehicle or other software linked to the digital signing process presents inaccurate data to the user, causing the user to digitally sign other data than the intended information. It is vital that the signing equipment present only the actual data to be signed. It would be a serious security issue if an EDL system presented to a signer data which does not correlate to the actual data being signed.

To guarantee a secure EDL framework, the digital signing equipment must be designed in a way that makes tampering impossible. One possibility would be a close coupling of the signing equipment and the display presenting the data to be signed to the user. Preventing the data to be signed from being sent across different hardware or software would reduce the potential for external manipulation.

For example, a malicious person might plant some data to be signed within a vehicle system, but the display would encourage the unknowing driver to sign some other information. Driver $d_1$ digitally signs a message containing a drive request regarding vehicle $v_1$ at time $t_1$. But the EDL software and hardware has been manipulated to present information so that he or she thinks it is a drive request regarding vehicle $v_2$ at time $t_2$. If someone were to extract this digitally signed false request from the vehicle $v_2$ and at a later time insert it into vehicle $v_1$, the digitally signed message could be sent to a verification service to start the vehicle without $d_1$ intentionally signing the drive request or being present. The digital signing
software and hardware have to be securely implemented and constructed to guarantee that the information presented is the actual information that is to be signed by the license holder. Therefore it must be essential to implement an EDL framework that uses secure signature-creation devices according to EU’s definition when handling digital signing processes.
Chapter 10  
Summary

The presented road safety statistics have first and foremost provided proof of the need to prevent unlicensed drivers from driving vehicles. The presented data shows that government agencies responsible for the enforcement of the laws seem to need the help of technology. Police officers cannot control the driving licenses of every driver driving a vehicle. And if there were enough police officers to control almost every vehicle on the road, it would probably result in traffic chaos, as well as a bankrupt government when all these police officers collected their monthly salaries.

The next step was to investigate whether the legal system could accept and support an implementation of a unified EDL framework. The EU has presented a directive on driving licenses that should be implemented by every member state. This directive unites the member countries’ laws and regulations regarding the issuing of driving licenses and the right to drive specific vehicle categories. As these rules and regulations do not diverge over a quite large geographical area, the implementation of an EDL framework within the EU should be realistic. Much of the reliability of the presented EDL framework lies in the usage of PKI and especially digital signatures. The legal aspects of the usage of digital signatures are also set by the EU in another directive, in which we pointed out information indicating that a digital signature should be as valid as any other handmade signature. The directive states that “Member states shall ensure that an electronic signature is not denied legal effectiveness and admissibility as evidence in legal proceedings solely on the grounds that it is in electronic form”.

As the EU’s member countries, and Sweden as a primary example, have merged their driving license legislation and issued laws regarding the usage of digital signatures, an EDL framework implementation should, based on these conditions, be possible.

When it comes to the presented technology and form factor, it is pretty much settled that the ID-1 form factor should be used. This choice is based on two facts. First of all is its acceptance by license holders, because this is a commonly used form factor today, and second, it is determined by the EU’s directive on driving licenses that the form factor ID-1 must be used.

Previous research done by Fred Goldberg has already shown that contactless smart cards are a more suitable solution to prevent everyday wear and tear of the EDLs.

The contactless smart card EDL would preferably be access protected by biometric data such as fingerprint scanning. The use of a PIN or password to access stored private keys and other on-card data would most likely be unsuitable because passengers sit close to drivers and the secret PIN or password may be revealed.

The EDL framework should make use of X.509 digital certificates as the technology is widely accepted to guarantee validity throughout systems that use PKI asymmetric encryption. Asymmetric encryption using public- and private-key pairs provides the system with the means to guarantee integrity and authentication of the involved entities. All three presented major entities, the driver, the vehicle and the verification service, assume that they are communicating with someone or something that really is what it or he or she claims to be.

Modern vehicles are already pretty much controlled by on-board computers. The controller area network (CAN) provides the EDL framework with a well-defined communication standard to be able to communicate with vehicle computers and/or other equipment.

The presented EDL possibly provides a technology that would be able to merge a lot of information on the issued EDL. Whether this is desirable is left undecided within the scope of this report. Both advantages and disadvantages are discussed when it comes to the merging of driving license and personal information on the EDL. But it is a possibility, and designers would have to clearly analyse data
access matters regarding different types of information. If designers merge this information on the EDL, it should be mandatory to utilize different access levels to the information. Perhaps a pure license information solution would result in a higher degree of personal privacy as a result of not including other personal information on the EDL.

In regions where vehicle network coverage is sufficient, the approach of online/offline adaptable EDL verification has to be the best solution. To implement an EDL framework that requires a permanent network connection is a hazardous approach that, in our present day, is bound to fail. Until we reach the point (if ever) when we have widespread permanent nonfailing vehicle network access, the online/offline approach has to be the best possible solution. This solution utilizes the best aspects of both online and offline access control, giving the designer adaptability regarding network coverage.

Finally, it is described how an EDL framework could result in privacy intrusion and what other security risks the implementation of such framework would pose if it were misused or poorly designed. Another point of view is that a totally bulletproof system regarding misuse by unlicensed drivers would perhaps result in minimal privacy for all drivers integrated within the system. Do we want a society where our every step is monitored? Would people accept being monitored when it comes to traffic safety to save many thousand lives every year? Should the EDL framework be a partially secure system, guaranteeing that almost everybody driving a vehicle has a valid license, and should those who intentionally misuse or manipulate the system be pursued by police officers or other governmental control instead of by a totally safe EDL system that might reduce people’s privacy? Again it is, in many aspects, a choice between saving lives or preserving drivers’ privacy. The aspects may not be black and white, but they should be analysed thoroughly and included within the system design to gain user acceptance.

Another aspect of trust is who we trust and whether they can be trusted. A good example of one such entity is a government. Can we trust a government and who is in control of the government? Governments are controlled by people, and as much as we would like to see a government as a trustworthy mechanical entity automatically striving towards a greater society for all its people, it is actually controlled by regular people, hopefully through laws and regulations given by the same people. But who ensures that the people controlling the government follow those laws and regulations? And who controls the controllers? Whom one can or should trust is hard to define. An EDL framework relies greatly on the fact that the governing authority can be trusted. But can governments in every country be trusted? History has revealed many cases when governments clearly have shown examples of untrustworthiness throughout the world.

We cannot value human lives in terms of economic resources, but everything has a cost, and the price tag of implementing an EDL framework has to be considered at some point. The economic aspect is beyond the scope of this report, but some would perhaps vote in favour of rather giving funds to other traffic safety–increasing measures or to increased traffic police work. One question is whether these funds would result in the same increase in traffic road safety if used somewhere else.

Finally, we can determine that technology has the potential to save thousands of lives by improving traffic safety. This increased traffic safety would in many cases be a result of increased control mechanisms regarding license rules and regulations. Designers and deploying authorities have to consider the pros and cons regarding traffic safety versus license holders’ privacy to implement a suitable option. Relatives and families of those seriously hurt or killed by unlicensed drivers would probably not protest increasing control mechanisms.

To save human lives and preventing people from getting hurt must be some of the best reasons for introducing an EDL framework and making it a part of our everyday life.

**10.1 Report results and influence on future development**

This report presents the design of an EDL framework consisting of an XML format EDL and an online and/or offline driver and vehicle verification service. The presented EDL framework is designed using
acknowledged technology and furthermore shows that EDLs are possible to integrate into today’s traffic environment. It is furthermore shown that technology can help us reach goals regarding increased traffic safety and that the existing laws and regulations do provide us with the administrative tools to deploy an EDL framework.

The report presents the physical format and layout as well as the digital content of an EDL. The presented XML format license also shows how information from multiple other documents could be merged and included into an EDL.

The design of the vehicle and verification service includes flow charts and data sources required for verifying EDLs in an online and/or offline approach.

Examples of EDL framework usage and some key issues regarding privacy are presented and discussed.

This report presents the possibility to implement and deploy an EDL framework that has the potential of saving many thousand lives and preventing even more people from being seriously hurt in traffic accidents. It brings together technology, legal aspects, statistical information, privacy and usage discussions into a pool of information required to implement future EDL frameworks.
Chapter 11 References


Chapter 12 Appendix I

12.1 EDL XML (EF_{EDL})

Separated elementary file implementation and elementary file names shown within parenthesis.

For example would EF_{EDL, DNR} be implemented as (in a separated elementary file stored on the EDL smart card):

```xml
<?xml version="1.0" encoding="utf-8"?>
<EDL>
  <DocumentNumber>0123456789</DocumentNumber>
</EDL>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<EDL>
  <DocumentNumber>0123456789</DocumentNumber>
  <IssuingDate>2014-10-29</IssuingDate>
  <ExpiryDate>2024-10-29</ExpiryDate>
  <IssuingAuthority>
    <Name>RoadAdministration</Name>
    <Address>
      <Street>Royal Street 1</Street>
      <PostalCode>543 21</PostalCode>
      <City>Umeå</City>
      <State/>
      <StateName />
      <StateCode />
    </State>
    <Country>
      <CountryName>Sweden</CountryName>
      <CountryCode>SE</CountryCode>
    </Country>
    <Latitude>63.825847</Latitude>
    <Longitude>20.263035</Longitude>
  </Address>
  <Telephone>(+46)701234567</Telephone>
  <Web>http://www.theissuingauthority.se</Web>
  <Email>info@theissuingauthority.se</Email>
</IssuingAuthority>
</EDL>

(EDL_DNR)
(EDL_Idate)
(EDL_Edate)
(EDL_Iauth)
(EDL_ICE)
EDL – Electronic Driving Licenses

</Person>

</Company>

</ICE>

</Identification>

</Biometrics>

</Identification>

</Address>

</ICE>

</Identification>

</Address>
<State>
<Latitude>64.749283</Latitude>
<Longitude>20.951368</Longitude>
</Address>
<DrivingLicense>
<LicenseNumber>1122344455</LicenseNumber>
<Categories>
<AM>
<Holder>true</Holder>
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</AM>
<A1>
<Holder>false</Holder>
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</A1>
<A2>
<Holder>false</Holder>
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</A2>
<A>
<Holder>false</Holder>
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</A>
<B1>
<Holder>false</Holder>
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</B1>
<B>
<Holder>true</Holder>
<IssuingDate>1999-11-24</IssuingDate>
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</B>
</Categories>
</DrivingLicense>
<IssuingDate />
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</C1>

<C>
  <Holder>false</Holder>
  <IssuingDate />
  <ExpiryDate />
  <Restrictions />
  <AdditionalInfo />
</C>

<D1>
  <Holder>false</Holder>
  <IssuingDate />
  <ExpiryDate />
  <Restrictions />
  <AdditionalInfo />
</D1>

<D>
  <Holder>true</Holder>
  <IssuingDate>2014-10-29</IssuingDate>
  <ExpiryDate>2024-10-29</ExpiryDate>
  <Restrictions>01.06</Restrictions>
  <AdditionalInfo />
</D>

<BE>
  <Holder>false</Holder>
  <IssuingDate />
  <ExpiryDate />
  <Restrictions />
  <AdditionalInfo />
</BE>

<C1E>
  <Holder>false</Holder>
  <IssuingDate />
  <ExpiryDate />
  <Restrictions />
  <AdditionalInfo />
</C1E>

<CE>
  <Holder>false</Holder>
  <IssuingDate />
  <ExpiryDate />
  <Restrictions />
  <AdditionalInfo />
</CE>

<D1E>
  <Holder>false</Holder>
  <IssuingDate />
  <ExpiryDate />
<Restrictions />
<AdditionalInfo />
</D1E>
<DE>
<Holder>false</Holder>
<IssuingDate />
<ExpiryDate />
<Restrictions />
<AdditionalInfo />
</DE>
</Categories>
</DrivingLicense>

<MedicalInformation>
<OrganDonor>
<AllOrgansAndTissue>false</AllOrgansAndTissue>
<SomeOrgansOrTissue>true</SomeOrgansOrTissue>
<Details>Not my left hand!</Details>
<StartDate>2014-10-28</StartDate>
<ExpiryDate>2024-10-28</ExpiryDate>
<DigitalSignature>(Signature of a witness)</DigitalSignature>
</OrganDonor>
</MedicalInformation>

<EuropeanHealthInsurance>
<CardIdentificationNumber>01234567890123456789</CardIdentificationNumber>
<Country>
<CountryName>Sweden</CountryName>
<CountryCode>SE</CountryCode>
</Country>
</EuropeanHealthInsurance>

<MedicalConditions>Alzheimer’s disease</MedicalConditions>
<Allergies>Cat allergy</Allergies>

<Medication>
{Name>Ibuprofen</Name>
<Dosage>800 mg orally</Dosage>
<Frequency>2 times per month</Frequency>
</Medication>

<ActionInCaseOfEmergency>Talk to me gently!</ActionInCaseOfEmergency>
</MedicalInformation>

<TrafficInsurance>
<InsuranceIdentificationNumber>123456789</InsuranceIdentificationNumber>
<InsuranceCompany>
{Name>Folksam</Name>
</InsuranceCompany>
</TrafficInsurance>
12.2 EDL XML schema

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="EDL">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DocumentNumber" type="xs:int"/>
        <xs:element name="IssuingDate" type="xs:date"/>
        <xs:element name="ExpiryDate" type="xs:date"/>
        <xs:element name="IssuingAuthority">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string"/>
              <xs:element name="Address">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Street" type="xs:string"/>
                    <xs:element name="PostalCode" type="xs:string"/>
                    <xs:element name="City" type="xs:string"/>
                    <xs:element name="State">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="StateName"/>
                          <xs:element name="StateCode"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
<xs:element name="Country">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CountryName" type="xs:string"></xs:element>
      <xs:element name="CountryCode" type="xs:string"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Latitude" type="xs:double"></xs:element>
<xs:element name="Longitude" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Telephone" type="xs:string"></xs:element>
<xs:element name="Web" type="xs:string"></xs:element>
<xs:element name="Email" type="xs:string"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ICE">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Person">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="xs:string"></xs:element>
            <xs:element name="Telephone" type="xs:string"></xs:element>
            <xs:element name="Email" type="xs:string"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Company">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="xs:string"></xs:element>
            <xs:element name="Telephone" type="xs:string"></xs:element>
            <xs:element name="Email" type="xs:string"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Identification">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Biometrics">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PhotographyLink" type="xs:string"></xs:element>
            <xs:element name="SignatureLink" type="xs:string"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="FingerprintLink" type="xs:string"></xs:element>
<xs:element name="PalmprintLink" type="xs:string"></xs:element>
<xs:element name="DnaLink" type="xs:string"></xs:element>
<xs:element name="RetinaLink" type="xs:string"></xs:element>
<xs:element name="IrisLink" type="xs:string"></xs:element></xs:sequence></xs:complexType></xs:element>
<xs:element name="PersonalIdentificationNumber" type="xs:int"></xs:element>
<xs:element name="Surname" type="xs:string"></xs:element>
<xs:element name="GivenNames" type="xs:string"></xs:element>
<xs:element name="Gender" type="xs:string"></xs:element>
<xs:element name="Height" type="xs:int"></xs:element>
<xs:element name="DateOfBirth" type="xs:date"></xs:element>
<xs:element name="CountryOfBirth">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CountryName" type="xs:string"></xs:element>
      <xs:element name="CountryCode" type="xs:string"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CountryOfCitizenship">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CountryName" type="xs:string"></xs:element>
      <xs:element name="CountryCode" type="xs:string"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SecondCountryOfCitizenship">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CountryName" type="xs:string"></xs:element>
      <xs:element name="CountryCode" type="xs:string"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PreferredLanguage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="LanguageName" type="xs:string"></xs:element>
      <xs:element name="LanguageCode" type="xs:string"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Address">
  <xs:complexType>
<xs:sequence>
  <xs:element name="Street" type="xs:string"></xs:element>
  <xs:element name="PostalCode" type="xs:string"></xs:element>
  <xs:element name="City" type="xs:string"></xs:element>
  <xs:element name="Country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="CountryName" type="xs:string"></xs:element>
        <xs:element name="CountryCode" type="xs:string"></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="State">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="StateName"></xs:element>
        <xs:element name="StateCode"></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Latitude" type="xs:double"></xs:element>
  <xs:element name="Longitude" type="xs:double"></xs:element>
</xs:sequence>

<xs:element name="DrivingLicense">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="LicenseNumber" type="xs:int"></xs:element>
      <xs:element name="Categories">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AM">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Holder" type="xs:string"></xs:element>
                  <xs:element name="IssuingDate"></xs:element>
                  <xs:element name="ExpiryDate"></xs:element>
                  <xs:element name="Restrictions"></xs:element>
                  <xs:element name="AdditionalInfo"></xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="A1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Holder" type="xs:string"></xs:element>
                  <xs:element name="IssuingDate"></xs:element>
                  <xs:element name="ExpiryDate"></xs:element>
                  <xs:element name="Restrictions"></xs:element>
                  <xs:element name="AdditionalInfo"></xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="IssuingDate"/>
<xs:element name="ExpiryDate"/>
<xs:element name="Restrictions"/>
<xs:element name="AdditionalInfo"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="C">
<xs:complexType>
<xs:sequence>
<xs:element name="Holder" type="xs:string"/>
<xs:element name="IssuingDate"/>
<xs:element name="ExpiryDate"/>
<xs:element name="Restrictions"/>
<xs:element name="AdditionalInfo"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="D1">
<xs:complexType>
<xs:sequence>
<xs:element name="Holder" type="xs:string"/>
<xs:element name="IssuingDate"/>
<xs:element name="ExpiryDate"/>
<xs:element name="Restrictions"/>
<xs:element name="AdditionalInfo"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="D">
<xs:complexType>
<xs:sequence>
<xs:element name="Holder" type="xs:string"/>
<xs:element name="IssuingDate" type="xs:date"/>
<xs:element name="ExpiryDate" type="xs:date"/>
<xs:element name="Restrictions" type="xs:double"/>
<xs:element name="AdditionalInfo" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="BE">
<xs:complexType>
<xs:sequence>
<xs:element name="Holder" type="xs:string"/>
<xs:element name="IssuingDate"/>
<xs:element name="ExpiryDate"/>
<xs:element name="Restrictions"/>
<xs:element name="AdditionalInfo"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="BE">
<xs:complexType>
<xs:sequence>
<xs:element name="Holder" type="xs:string"/>
<xs:element name="IssuingDate"/>
<xs:element name="ExpiryDate"/>
<xs:element name="Restrictions"/>
<xs:element name="AdditionalInfo"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="C1E">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Holder" type="xs:string"></xs:element>
      <xs:element name="IssuingDate"></xs:element>
      <xs:element name="ExpiryDate"></xs:element>
      <xs:element name="Restrictions"></xs:element>
      <xs:element name="AdditionalInfo"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CE">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Holder" type="xs:string"></xs:element>
      <xs:element name="IssuingDate"></xs:element>
      <xs:element name="ExpiryDate"></xs:element>
      <xs:element name="Restrictions"></xs:element>
      <xs:element name="AdditionalInfo"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="D1E">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Holder" type="xs:string"></xs:element>
      <xs:element name="IssuingDate"></xs:element>
      <xs:element name="ExpiryDate"></xs:element>
      <xs:element name="Restrictions"></xs:element>
      <xs:element name="AdditionalInfo"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="DE">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Holder" type="xs:string"></xs:element>
      <xs:element name="IssuingDate"></xs:element>
      <xs:element name="ExpiryDate"></xs:element>
      <xs:element name="Restrictions"></xs:element>
      <xs:element name="AdditionalInfo"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="MedicalInformation"/>
12.1 Additional driving license information and restriction(s)

The European Union has stated that additional information and restrictions could be included in the issued driving licenses.

The driving license additional information and restriction is defined by the European Union (the European Union, 2006) as number codes separated by dots. According to the directive the following group of information and restriction codes will be used throughout Europe:

(Code groups 1–99 are harmonized European Union codes, used in every member state implementing the driving license directive.)

**DRIVER** (Medical reasons)

01. Sight correction and/or protection
02. Hearing aid/communication aid
03. Prosthesis/orthosis for the limbs
05. Limited use (subcode use obligatory, driving subject to restrictions for medical reasons)

**VEHICLE ADAPTATIONS**

10. Modified transmission
15. Modified clutch
20. Modified braking systems
25. Modified accelerator systems
30. Modified combined braking and accelerator systems
35. Modified control layouts
40. Modified steering
42. Modified rear-view mirror(s)
43. Modified driver seat
44. Modifications to motorcycles (subcode use obligatory)
45. Motorcycle with side-car only
46. Tricycles only
50. Restricted to a specific vehicle/chassis number (vehicle identification number, VIN)
51. Restricted to a specific vehicle/registration plate (vehicle registration number, VRN)

**ADMINISTRATIVE MATTERS**

70. Exchange of recognized driving license
71. Duplicate of license/permit
73. Restricted to category B vehicles of the motor quadricycle type (B1)
78. Restricted to vehicles with automatic transmission
79. License category restriction
80. Restricted to holders of a driving license for a category A vehicle of the motor tricycle type not having reached the age of 24 years

81. Restricted to holders of a driving license for a category A vehicle of the two-wheel motorcycle type not having reached the age of 21 years

90. Codes used in combination with codes defining modifications of the vehicle

96. Can tow a trailer whose authorized weight does not exceed 4,250kg

97. Not authorized to drive a category C1 vehicle which falls within the scope of Council Regulation (EEC) No 3821/85 of 20 December 1985 on recording equipment in road transport (1)

(Code groups 100 and above are national codes valid only for driving in the territory of the member state which issued the license.)

Some of the listed categories have subcategories. An example of a commonly used restriction code of this type is

01.06. A restriction that requires the driver to wear glasses or contact lenses when he or she drives the vehicle.