Kerstin Lemke
Christof Paar
Marko Wolf (Eds.)

Embedded Security in Cars

Securing Current and Future Automotive IT Applications

Springer

Embedded Security in Cars

Kerstin Lemke · Christof Paar · Marko Wolf (Eds.)

Embedded Security in Cars

Securing Current and Future Automotive IT Applications

With 53 Figures and 25 Tables

Springer

Editors

Kerstin Lemke
Ruhr-Universität Bochum
44780 Bochum, Germany
lemke@crypto.rub.de
www.crypto.rub.de

Marko Wolf
Ruhr-Universität Bochum
44780 Bochum, Germany
mwolf@crypto.rub.de
www.crypto.rub.de

Christof Paar
Ruhr-Universität Bochum
44780 Bochum, Germany
cpaar@crypto.rub.de
www.crypto.rub.de

Library of Congress Control Number: 2005935329

ACM Computing Classification (1998): C.3, C.5, E.3, J.7

ISBN-10 3-540-28384-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28384-3 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typeset by the authors using a Springer TEX macro package
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Cover design: KünkelLopka Werbeagentur, Heidelberg

Printed on acid-free paper 45/3142/YL - 5 4 3 2 1 0

Preface

Information technology is the driving force behind innovations in the automotive industry, with perhaps 90% of all innovations in cars based on electronics and software. Up to 80 embedded processors can be found in a high-end car, and electronics and software are already a major cost factor in car manufacturing. The situation is similar for commercial vehicles such as trucks. One crucial aspect of future IT applications in vehicles is the security of these systems. Whereas software safety is a relatively well-established (if not necessarily well-understood) field, the protection of automotive IT systems against manipulation has only very recently started to emerge. When we started working in this exciting area about four years ago, we realized that there is hardly any literature on this topic, not to mention any kind of comprehensive description of the field of IT security in cars.

This book has a simple main objective: *We attempt to give an overview on most aspects which are relevant for IT security in automotive applications.* We hope that the book is, on the one hand, of interest to automotive engineers and technical managers who want to learn about security technologies, and, on the other hand, for people with a security background who want to learn about security issues in modern automotive applications. In particular, we hope that the book can serve as an aid for people who need to make informed decisions about car security solutions, and for people who are interested in research and development in this exciting field.

As can be seen from the table of contents, IT security in cars incorporates quite diverse disciplines. In addition to its spread across different technical areas, it is a new and fast-moving field, so that the collection of topics in this book should be viewed as a "best guess" rather than the final word on what exactly constitutes automotive IT security. All of the contributing authors (and ourselves) have been working for many years in embedded security, and for a few years on various aspects of car security from a research as well as from an industry viewpoint.

The book consists of an introduction and three other main parts. The first article, *Embedded IT Security in Automotive Application – an Emerging*

Area, provides an overview of the field and at the same time serves as an introduction and motivation for the remainder of this book.

Part II, Security in the Automotive Domain, is a collection of articles which describe the most relevant **car applications for which IT security is crucial**. The range of topics is quite broad, including security for immobilizers, tachographs, software updates ("flashing"), communication buses and vehicle communication. Some of the topics are very current, such as secure flashing, whereas other topics such as inter-vehicle communication are forward looking.

Part III, Embedded Information Technology in Cars: State-of-the-art, deals with the actual **security technologies** that are relevant for securing car applications. In each article a comprehensive introduction to important aspects of embedded security is given. The goal here was to inform in an understandable manner about topics such as current symmetric and asymmetric cryptography, physical security, side-channel attacks and wireless security. The articles attempt to provide the most important facts which can assist people with an automotive background without overloading the reader with too much theoretical detail.

Part IV, Business Aspects of IT Systems in Cars, shows the interdisciplinary dimension of IT security in the car context. The authors show in three separate articles that security is a central tool for novel **IT-based business models**. This part of the book is perhaps the one that demonstrates best the enormous impact that IT security has in cars, which goes well beyond a mere technical one.

We hope that the book is of interest to people in industry and academia, and also hope that it helps somewhat to enhance the field of embedded IT security in cars.

Bochum,
October 2005

Kerstin Lemke
Christof Paar
Marko Wolf

Contents

List of Contributors

André Adelsbach
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
andre.adelsbach@nds.rub.de

Dr. Igor Furgel
T-Systems GEI GmbH
Solution & Service Center
Test Factory & Security
Rabin Str. 8
53111 Bonn, Germany
igor.furgel@t-systems.com

Dr. Martin Gersch
Competence Center E-Commerce (CCEC)
Ruhr University of Bochum
44780 Bochum, Germany
martin.gersch@rub.de

Marcus Heitmann
Institute for E-Business Security (ISEB)
Ruhr University of Bochum
44780 Bochum, Germany
marcus.heitmann@volkswagen.de

Prof. Jean-Pierre Hubaux
School of Computer and Communication Sciences
EPFL
CH-1015 Lausanne, Switzerland
jean-pierre.hubaux@epfl.ch

Ulrich Huber
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
huber@crypto.rub.de

Sandeep Kumar
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
kumar@crypto.rub.de

Kerstin Lemke
Horst Görtz Institute for IT Security
Ruhr University of Bochum,
44780 Bochum, Germany
lemke@crypto.rub.de

Jun Luo
School of Computer and Communication Sciences
EPFL
CH-1015 Lausanne, Switzerland
jun.luo@epfl.ch

Markus Müller
T-Systems GEI GmbH
Solution & Service Center
Test Factory & Security
Rabin Str. 8
53111 Bonn, Germany
mmueller@t-systems.com

Prof. Christof Paar
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
cpaar@crypto.rub.de

Jan Pelzl
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
pelzl@crypto.rub.de

Solveig Richter
T-Systems GEI GmbH
Solution & Service Center
Test Factory & Security
Rabin Str. 8
53111 Bonn, Germany
solveig.richter@t-systems.com

Klaus Rüdiger
Institute for E-Business Security (ISEB)
Ruhr University of Bochum
44780 Bochum, Germany
klaus.ruediger@rub.de

Prof. Ahmad-Reza Sadeghi
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
sadeghi@crypto.rub.de

Kai Schramm
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
schramm@crypto.rub.de

Winfried Stephan
T-Systems GEI GmbH
Solution & Service Center
Test Factory & Security
Rabin Str. 8
53111 Bonn, Germany
winfried.stephan@t-systems.com

Christian Stüble
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
stueble@crypto.rub.de

Dr. André Weimerskirch
escrypt GmbH – Embedded Security
Lise-Meitner-Allee 4
44801 Bochum, Germany
aweimerskirch@escrypt.com

Marko Wolf
Horst Görtz Institute for IT Security
Ruhr University of Bochum
44780 Bochum, Germany
mwolf@crypto.rub.de

Dr. Thomas Wollinger
escrypt GmbH – Embedded Security
Lise-Meitner-Allee 4
44801 Bochum, Germany
twollinger@escrypt.com

Part I

Introduction

Embedded IT Security in Automotive Application – An Emerging Area

Christof Paar

Horst Görtz Institute (HGI) for IT Security
Ruhr University of Bochum, Germany
cpaar@crypto.rub.de

Summary. Information technology has gained central importance for many new automotive applications and services. The majority of innovation in cars is based on software and electronics, and IT-related costs are approaching the 50% margin in car manufacturing. We argue that security will be a central technology for the next generation of automobiles. We list application domains, i.e., whole families of automotive applications, which rely heavily on IT security. This article also introduces core security technologies relevant for car systems. It is explained that embedded security, as opposed to general IT security, is highly relevant for car applications. The article concludes with specific recommendations for the successful introduction of security solutions in automotive applications.

1 Introduction

Information technology – we define broadly as being systems based on digital hardware and software – has gained central importance for many new automotive applications and services. On the production side we observe that the cost for electronics and IT is approaching the 50% threshold of all manufacturing costs. Perhaps more importantly, there are estimates that already today more than 90% of all vehicle innovations are centered around software and hardware (admittedly not only digital hardware, though). IT systems in cars can roughly be classified into three main areas:

1. Basic car functions, e.g., engine control, steering, and braking.
2. Secondary car functions, e.g., window control and immobilizers.
3. Infotainment applications, e.g., navigation systems, music and video entertainment, and location-based services.

For a good overview on the possible future of car IT systems, the reader is referred to [3].

Almost all such applications are realized as embedded systems, that is, as *devices* which incorporate a microprocessor. The devices range from simple

control units based on an 8-bit micro controllers to infotainment systems equipped with high-end processors whose computing power approaches that of current PCs. The number of processors can be 80 or more in high-end cars. In a typical automobile the devices are connected by several separate buses.

Not surprisingly, many classical IT and software technologies are already well established within the automotive industry, for instance hardware-software co-design, software engineering, software component re-use, and software safety. However, one aspect of modern IT systems has hardly been addressed in the context of automotive applications: IT security. Security is concerned with protection against the manipulation of IT systems by humans. The difference between IT safety and security is depicted in Fig. 1[1].

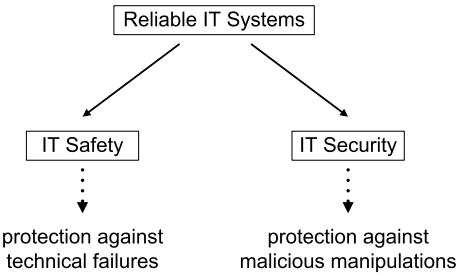

Fig. 1. The relationship between IT safety and security

As said above, software and hardware safety is a relatively well-established (if not necessarily well-understood) field in the automotive industry, IT security, on the other hand, is just beginning to emerge as a proper sub-discipline within the field of automotive IT. Of course, there have been niche applications in the automotive domain, especially concerned with electronic immobilizers, that have always relied on security technologies. However, the vast majority of software and hardware systems in current cars are not equipped with security functionality. This is not entirely surprising for two reasons:

1. Many past car IT systems did not need security functions as there was very little incentive for malicious manipulation in traditional applications.
2. Security tends to be an afterthought in any IT system. Achieving the core function, i.e., getting a telematic system working or enabling remote software updates, is the primary goal of every system designer and implementer. A prime example of such an IT-system is the Internet, which is only now, after several decades of existence, being equipped with rudimentary security functions.

[1] In German it is especially easy to confuse the two terms, since both safety and security translate into the same word: *Sicherheit*.

As we will see in the remainder of this contribution – and in much more detailed in the other articles of this volume – the situation has changed dramatically with respect to the first argument given above. Already today there is a multitude of quite different car sub-systems that are in desperate need for strong security functions in order to protect the driver, the manufacturer and the component supplier. Current examples of car functions with need for security include the large field of software updates, also known as "flashing" or "chip tuning". Future cars will become even more dependent on IT security due to the following developments:

- It is predicted that an increasing number of ECUs (electronic control units) will be reprogrammable, a process that must be protected.
- Many cars will communicate with the environment in a wireless fashion, which makes strong security a necessity.
- New business models (e.g., time-limited flash images or pay-per-use infotainment content) will become possible for the car industry, but will only be successful if abuse can be prevented.
- There will be an increasing number of legislative demands which can only be solved by means of modern IT security functions, such as tamper-resistant tachographs, secure emergency call functions, secure road billing etc.
- Increasing networking of cars will allow the collection of data for each driver (e.g., driving behavior, locations visited), which will put high demands on privacy technology.
- Future cars will often be personalized, which requires a secure identification of the driver.
- Electronic anti-theft measures will go beyond current immobilizers, e.g., by protecting individual components.

As we can see from the, not necessarily complete, list above, IT security will be an important topic for many future car technologies. For some future applications, such as business models based on Digital Rights Management, IT security will even play the role of an enabling technology.

We would like to stress at this point that almost all target platforms within cars which will incorporate security functions are embedded systems, rather than classical PC-style computers. Hence, the technologies needed for securing car applications belong often, but not always, to the field of embedded security. The difference between embedded security vs. general IT security will be discussed in more detail in Section 3. A good introduction to embedded security is give in [1].

2 Automotive Applications and IT Security?

As sketched above, embedded IT security will be a crucial part of many future automotive features. IT security offers a wide variety of functions that

can improve products. In the context of embedded automotive systems, the advantages of strong IT security can be summarized in two main categories.

- **Increased reliability**: Innovative IT applications must be protected against targeted manipulations. For instance, manipulation of an otherwise robust electronic engine control system may result in an unreliable engine (e.g., shortened engine life span). Another example is a highly fault-tolerant telematic system. Manipulation of messages to and from the car, however, may result in a very unreliable system. IT security can prevent those and many other abuses. It is important to stress that from this viewpoint *security can also be interpreted as being part of reliability.*
- **New business models**: Cars equipped with state-of-the-art IT technology will open up opportunities for a multitude of new business models. In times where international competition is putting increasing pressure on car manufacturers, novel IT-based business models are tempting options. Examples include fee-based software updates, navigation data, location-based services and multimedia content. It is of crucial importance to stress that virtually all such business models rely heavily on strong IT security.

Admittedly, this is a rather broad classification. In the following we will list more concrete application domains within cars that rely heavily on IT security.

Software Updates. In the last few years the topic of software updates of ECUs (electronic control units) has gained crucial importance. The reasons why ECUs that can be updated are attractive are multitude, and a few important ones are given in the following: many software bugs are only found after shipment of the car; cars can be configured differently for different customers, reducing the variety of cars that have to be manufactured; features can be activated based on a pricing policy. Unfortunately, unauthorized software updates can pose an equal number of problems for manufactures and, to a lesser degree, for owners. For instance, it is obviously quite attractive to activate certain car features (e.g., a stronger engine or comfort functions) without paying the associated fees. This cannot only result in financial losses due to missed business transactions, but also in an increase of warranty cases. In order to enable software update in a manner controlled by the manufacturer, one needs embedded security technologies such as digital signature, tamper-resistant hardware and encryption. The articles *Secure Software Updates* and *Secure Software Delivery and Installation in Embedded Systems* in this volume deal with the topic in detail.

Theft Prevention. The electronic immobilizer is possibly the oldest incarnation of IT security mechanisms in the automotive. As documented in [12], the latest generation of immobilizer has been quite a success, with the damage from car thefts reduced by 50% over the last decade or so. This proves that classical IT security (here: strong cryptographic identification) can have an immediate benefit in today's world. It is very tempting to generalize immobilizer solutions to car components. By using strong cryptography, one could

identify valuable or crucial components and, thus, protect against illegal exchange of components, and enforce the usage of original manufacturer spare parts. The techniques required from embedded security are identification protocols and tamper resistance.

Business Models for Infotainment Content. It seems almost certain that the majority of future cars will be equipped with powerful infotainment devices in the dashboard. The functionality of the infotainment systems will be a fusion of

- home entertainment (e.g., radio, music and video for the rear seats),
- telecommunication (e.g., cell phone and email function),
- car-specific information systems (e.g., navigation data, smart traffic routing, emergency calls).

(See also the article *Security Risks and Business Opportunities in In-Car Entertainment* in this volume for a more detailed discussion of this topic.) There will be opportunities for content providers, car manufacturers and possibly for other parties to create innovative business models around the digital content mentioned above. There are already systems in use today which provide navigation data on a time-limited basis, as explained in the article *In-vehicle M-Commerce: Business Models for Navigation Systems and Location-Based Services* in this volume. Another indication for the opportunities ahead is that in 2004 more than 50% of all mini vans sold in the USA were equipped with rear seat video screens. Adding new business models, for instance fee-based video downloads at hot spots, seems not entirely unrealistic.

The topic of security plays a crucial role here. It should be noted that there is a built-in incentive for users (i.e., business partners!) to behave dishonestly, e.g., by copying content in an unauthorized manner or by using content beyond the paid-for period. This situation is similar to the hotly debated topic of content distribution via the Internet. In order to prevent abuse and, thus, to enable new business models, strong embedded security technologies are needed. First, communication security (i.e., protection of the link between car and the environment) is needed in order to transport valuable digital content to the customer. Second, digital rights management (DRM) technologies are required to prevent the customer from unauthorized copying or an unauthorized extension of the usage period of the content. Third, privacy-preventing technology will be required in order to limit the collection of customer data. Without the latter measure, user acceptance of new technologies can quickly diminish. Finally, secure hardware components are required in order to prevent manipulation of the IT security mechanisms and demolishing the business model.

Personalization of Cars. Car functions that can be updated open up a wide variety of new possibilities such as personalization of car features, from your favorite radio station and seat position to your favorite suspension setting. There is a multitude of options for realizing a recognition of the driver. One class of approaches is token-based, e.g., through car keys, smart cards, or

cell phones. Other approaches make use of biometrics, e.g, fingerprint recognition. Another class simply requires active user input in order to communicate the person's identity to the vehicle. Again, we will need security technologies here in order to prevent abuse. Technologies needed included identification techniques, biometrics and tamper resistance hardware.

Access Control for Car Data. Already today many cars are equipped with event data recorders. This can be as simple as tachograph data, or more advanced systems which record a wide variety of information about the car subsystems and driving behavior. Currently, such data can usually only be accessed via diagnosis interfaces which have to be attached physically to the car. However, it seems likely that many vehicles will be equipped with wireless interfaces such as Bluetooth or GSM in the future. It becomes crucial now to tightly control access to both technical data about the car and stored information about driving behavior. Relevant security functions are authentication and identification protocols and communication security. The article *Security Aspects of Mobile Communication Systems* deals with security topics for wireless protocols.

Anonymity. Cars filled with IT systems offer several possibilities for violating drivers' privacy rights. The above-mentioned recording of driving behavior is one example. Navigation data used or requests for other location-based services (e.g., the purchase of certain navigation data, requests for the nearest gas station or requests for the nearest restaurant) is another example. It is also imaginable that even traffic violations, e.g., driving beyond the allowed speed limit, are recorded. These can all be serious threats in an information society and it will be crucial to prevent abuses by incorporating technologies such as access control and anonymization.

Legal Obligations. Already today there are several regulations that dictate the inclusion of IT security functions in cars. An example is *Toll Collect*, the German road toll system or the European tachograph, as described in the contribution *A Review of the Digital Tachograph System* in this volume. In the future, there will be more applications which require IT security due to legal regulations. Possible examples include emergency call systems, immobilizers or other theft control measures, and event data recorders.

We do not claim that the listing above is complete. However, we believe that embedded security is already an important technology for a host of diverse car functions, and its impact will increase in the future. In summary, it can be claimed that IT security will play the role of an enabling technology for numerous future car applications.

3 Embedded Security Technologies in Vehicles

3.1 Embedded Security vs. General IT Security

Since the late 1990s embedded security, sometimes also referred to as security engineering or cryptographic engineering, has emerged as a proper subdisci-

pline within the security and cryptography communities. Embedded security is often quite different from the security problems encountered in computer networks such as LANs or the Internet. For such classical networks there exist established and relatively mature security solutions, e.g., firewalls, encryption software, and intrusion detection systems. The topics with which embedded security deals are, generally speaking, closer related to the underlying software and hardware of the target device which is to be protected. Arguably the most important event at which embedded security technologies are treated from a scientific viewpoint is the CHES (Cryptographic Hardware and Embedded Systems) Workshop series [4, 5, 6, 7, 8, 9, 10] which started in 1999.

Even though there are certainly many aspects of security that are shared by embedded devices and general computers, there are a number of key differences: First, embedded devices tend to have small processors (often 8-bit or 16-bit micro-controllers) which are limited with respect to computational capabilities, memory, and power consumption. Modern PCs, on the other hand, are very powerful and in most cases do not limit the use of cryptographic functions. Second, potential attackers of embedded systems have often access to the target device itself, e.g., an attack of a smart card only makes sense if one actually has physical control over the smart card. On the other hand, attacks against traditional computer networks are almost always performed remotely. Third, embedded systems are often relatively cheap and cost sensitive because they often involve high-volume products which are priced competitively. Thus, adding complex and costly security solutions is not acceptable. By comparing typical prices (e.g., a laptop vs. an ECU) one easily notices a ratio of 1–2 orders of magnitude which, of course, limits the costs that can be spent on security for embedded solutions.

3.2 Cryptographic Algorithms in Constraint Environments

Even though security depends on much more than just cryptographic algorithms – a robust overall security design including secure protocols and organizational measures are needed as well – crypto schemes are in most cases the atomic building blocks of a security solution. The problem in embedded applications is that they tend to be computationally and memory constrained due to cost reasons. (Often they are also power limited, but since automotive applications are often powered by their own battery, low-power crypto is not such an important topic in the car context.) It is now the task of the embedded security engineer to implement secure crypto algorithms on small devices at acceptable running times.

Crypto schemes are divided into two families: symmetric and asymmetric algorithms. The first group is mainly used for data encryption and message integrity checks. Symmetric algorithms tend to run relatively fast and often need little memory resources. There exists a wealth of established algorithms, with the most prominent representatives being the block ciphers DES (Data Encryption Standard) and AES (Advanced Encryption Standard.) The family

of stream ciphers can be even more efficient than block ciphers and are, thus, sometimes preferred for embedded applications. In almost all cases it is a wise choice to use established, proven algorithms rather than unproven or self-developed ones. More on the state-of-the-art of symmetric algorithms will be said in the contribution *Fundamentals of Symmetric Cryptography* of this volume.

The second family of schemes, asymmetric or public-key algorithms, are very different. They are based on hard number theoretical problems and involve complex mathematical computations with very long numbers, commonly in the range of 160–4048 bits, depending on the algorithm and security level. Their advantage, however, is that they offer advanced functions such as digital signatures and key distribution over unsecure channels. For common automotive applications such as secure flashing, public-key algorithms are often preferred. The problem here is the computational requirement of public-key schemes. Embedded processors in the automotive domain are often only equipped with 8-bit and 16-bit processors clocked at moderate frequencies of, say, below 10 MHz. Running computationally expensive public-key algorithms on such processors can result in unacceptably long execution times, for instance several seconds for the generation of a digital signature. For this reason, it is very important that a smart parameter choice together with the latest implementation techniques are being employed. Much more details about properties and the selection of public-key algorithms will be given in *Fundamentals of Asymmetric Cryptography* of this volume.

3.3 Physical Security: Side Channel Attacks and Reverse Engineering

A central tool for providing security are cryptographic algorithms. Both symmetric and asymmetric algorithms are based on the fact that the protected device (e.g., tachograph, an ECU, or an infotainment device) is equipped with a *secret* cryptographic key. "Secret" means in this context also that it can not be read out by an attacker. If an attacker obtains knowledge of the key, the device can usually be manipulated and/or cloned. Many of the potential attackers – which includes in particular the owner and maintenance personnel – have physical access to the device.

One family of attacks which attempt to recover the key from the device are side channel attacks, which were first proposed in the open literature in 1996 [11]. Side channel attacks observe the power consumption, the timing behavior or the electromagnetic radiation of an embedded device. These signals are recorded while the cryptographic algorithm with the secret key is executed. The attacker then tries to extract the key by means of signal processing techniques. Side channel attacks are a serious threat in the real world unless special countermeasures have been implemented. Much more about side channels will be said in the contribution *Embedded Cryptography: Side Channel Attacks* in this volume.

A related family of attacks are fault injection attacks, sometimes referred to as active side channel attacks. Fault injection attacks force the device to malfunction, for instance by spikes in the power supply, through overclocking, or through overheating of the embedded device. The goal is often to create an incorrect output of the cryptographic algorithm which leaks information about the key used.

Quite different from side channel and fault injection attacks are reverse engineering attacks. The goal here is to read the cryptographic keys directly from the RAM, EEPROM, FlashROM, or ROM of the embedded device. Unlike classical reverse engineering of code it is in this context sometimes sufficient to recover a single cryptographic key for a successful attack, which is often only 16 bytes long or less. Of course, there is tamper-resistant memory available but it is for automotive systems often not available for cost or legacy reasons. Case studies about tamper resistance in the real world are described in [2]. A more detailed treatment is given in the article *Embedded Security: Physical Protection against Tampering Attacks* in this volume.

3.4 Digital Rights Management (DRM)

DRM has become a very important technology for applications such as audio and film distribution over the Internet. DRM systems can enforce rules such as the time period for which access to a music file is granted or to which device a digital movie is allowed to be copied. It is perhaps a bit surprising that DRM should become important for vehicles as well. However, as soon as digital data used for car applications represent financial values, e.g., flash software, digital location-based services or entertainment content, DRM will be the technology that enforces the envisioned use of the data. DRM technologies are required to prevent the customer from unauthorized copying or an unauthorized extension of the usage period of the content.

In order to realize a proper DRM platform in a vehicle, we need trusted computing functions which in turn are based on physical secure components such as secure memory, true random number generators and cryptographic algorithms.

3.5 Further Topics

The topics discussed above are certainly not comprehensive. However, they form a core of embedded security technologies that are relevant for most security solutions in cars. Topics such as mobile security are also treated in this volume.

4 Conclusion: Challenges and Opportunities for the Automotive IT Community

In summary it can be stated that embedded IT security in cars:

1. protects against manipulations by outsiders, owners and maintenance personnel,
2. increases the reliability of a system,
3. enables new IT-based business models.

As sketched above, there are several difficulties to overcome in order to develop strong embedded security solutions. We would like to give an outlook on the future of IT security in cars in the form of the following recommendations and conclusions:

- IT security will be a necessary requirement for many future automotive applications.
- Security will allow a multitude of new IT-based business models, e.g., location-based services or fee-based flashing. For such systems, security will be an enabling technology.
- Security will be integrated invisibly in embedded devices. Embedded security technologies will be a field in which manufacturers and part suppliers need to develop expertise.
- Security solutions have to be designed extremely carefully. A single "minor" flaw in the system design can render the entire solution unsecure. This is quite different from engineering most other technical systems: a single non-optimum component usually does not invalidate the entire system. An example is the Content Scrambling System (CSS) for DVD content protection, which was broken easily once it was reverse engineered.
- Embedded security in vehicles has to deal with very specific boundary conditions: computationally and memory constrained processors, tight cost requirements, physical security.
- The multi-player manufacturing chain for modern vehicles (OEM and possibly several layers of suppliers) can have implications for the security design. It is, for instance, relevant who designs a security architecture and, most importantly, who has control over the cryptographic keys.
- Merging the automotive IT and the embedded security community will allow many new applications. However, there are also several challenges: security and cryptography has historically been a field dominated by theoreticians, whereas the automotive IT is usually done by engineers. The culture in those two communities is quite different at times, and both sides have to put effort into understanding each other's way of thinking and communicating.

References

1. R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley and Sons, 2001.
2. R. J. Anderson and M. G. Kuhn. Tamper Resistance – a Cautionary Note. In *Second Usenix Workshop on Electronic Commerce*, pages 1–11, November 1996.
3. Manfred Broy. Herausforderungen der sicherheitsrelevanten Software im Automobilbereich. Key Note presentation at escar 2004, Bochum, Germany, November 2004.
4. Ç. K. Koç and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES'99*, volume LNCS 1717, Berlin, Germany, 1999. Springer-Verlag.
5. Ç. K. Koç and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2000*, volume LNCS 1965, Berlin, Germany, 2000. Springer-Verlag.
6. Ç. K. Koç, D. Naccache, and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2001*, volume LNCS 2162, Berlin, Germany, 2001. Springer-Verlag.
7. B. S. Kaliski, Jr., Ç. K. Koç, and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2002*, volume LNCS 2523, Berlin, Germany, 2002. Springer-Verlag.
8. Ç. K. Koç, C. Paar, and C. Walter, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2003*, volume LNCS 2779, Berlin, Germany, 2003. Springer-Verlag.
9. M. Joye and J.-J. Quisquater, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, volume LNCS 3156, Berlin, Germany, 2004. Springer-Verlag.
10. J. R. Rao and B. Sunar, editors. *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2005*, volume LNCS 3659, Berlin, Germany, 2005. Springer-Verlag.
11. P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume LNCS 1109, pages 104–113. Springer-Verlag, 1996.
12. W. Thönnes and S. Kruse. Electronical driving authority – how safe is safe? *VDI Berichte*, (1789), 2003.

Part II

Security in the Automotive Domain

Aspects of Secure Vehicle Software Flashing

Winfried Stephan, Solveig Richter, and Markus Müller

T-Systems GEI GmbH
Solution & Service Center Testfactory & Security
Rabinstr. 8
53111 Bonn, Germany
{winfried.stephan, solveig.richter, mmueller}@t-systems.com

Summary. This paper generalizes the practical experience gained from several projects. Processes of flashing based on the presented considerations are already in practical use.

1 Introduction

The volume of in-vehicle electronics and software integrated into today's vehicles has been increasing significantly for some time. Large numbers of sensors continuously provide a huge amount of data for a variety of measurement categories, e.g., concerning the vehicle status. This information must be analyzed electronically in real time. A growing number of in-vehicle functions is implemented and controlled by embedded systems. The number of electronic control units (ECUs) in modern vehicles averages 40 in the compact class and tops 90 in the luxury class.

By now, electronics have invaded virtually all vehicle functions. Examples of electronic-control aggregates are motor and gearbox control units in power transmission or servo steering, electrical window lift and climate control in the area of comfort electronics.

All of these functions are generally controlled by embedded systems. An *embedded system* is a specialized computer system that is part of a larger system or machine. Typically, an embedded system is housed on a single microprocessor board with the programs stored in ROM. Virtually all appliances that have a digital interface – watches, microwaves, cars – utilize embedded systems. Some embedded systems include an operating system but many are so specialized that the entire logic can be implemented as a single program.

In addition, modern vehicles are equipped with ECUs for safety functions such as ABS, ESP, airbag and in the infotainment field with systems such as navigation, broadcast, DVD players, and hands-free sets.

Embedded systems used for such functions today are generally provided with programmable flash memory instead of the fixed ROM modules used

earlier. This allows for the repair of software bugs in ECUs by flashing of new software versions instead of replacing the complete ECU unit. Another advantage thereby achieved is a reduction in the number of hardware variants for any one ECU type. In both cases, this results in a considerable cost benefit.

Flash memories also enable the integration of additional functionality by flashing new software instead of fitting new ECUs. Thus, a new business case for automotive Overall Equipment Manufacturers (OEMs) is born: vending of software.

2 Trusted Flashing – a Challenge

The challenge faced by OEMs by the introduction of flashing in ECUs lies in the necessity of establishing a complete software delivery process including the involvement of many different parties with possibly conflicting interests.

Among them are component developers, including suppliers and OEMs, in-plant component experts, after-sales services as well as non-captive maintenance workshops. In flashing ECUs, all of these players have to be organized into a single process enabling and ensuring the introduction of a correct and up-to-date software version into an ECU at any time. Among other things the software delivery process has to take into account late software modifications for new vehicles at the end of line (end-of-line flashing) due to the integration of last-minute changes and corresponding challenges in service. In the following the challenges of the process of flashing in service will be looked at in detail.

Figure 1 displays the main and corollary processes of flashing.

The main processes (grey fields) include all those processes necessary to provide software to an ECU. Corollary processes (blue fields) include all preaged activities, i.e., the development of the ECU hardware and the roll-out for the logistic and service processes.

The main processes consist of software (SW) development, followed by provisioning, distribution and finally flashing. After development and successful testing, the release of the software and its provision in service will take place. One of the biggest challenges is the detection of the conditions under which the flashware has to be released and the presentation of these conditions to the service. The conditions include, for example, the current hardware and software configuration of the vehicle. Therefore not only the ECU hardware but possibly also the configuration of the entire vehicle has to be documented.

In software distribution a very heterogeneous information network has to be assumed. Today, the distribution of software is generally effected by distributing CDs. Also, networks like the Internet or an OEM's intranet are used to transfer software to the services. At present, the flashware is usually brought into the ECUs by using special service equipment, e.g. diagnostic tester.

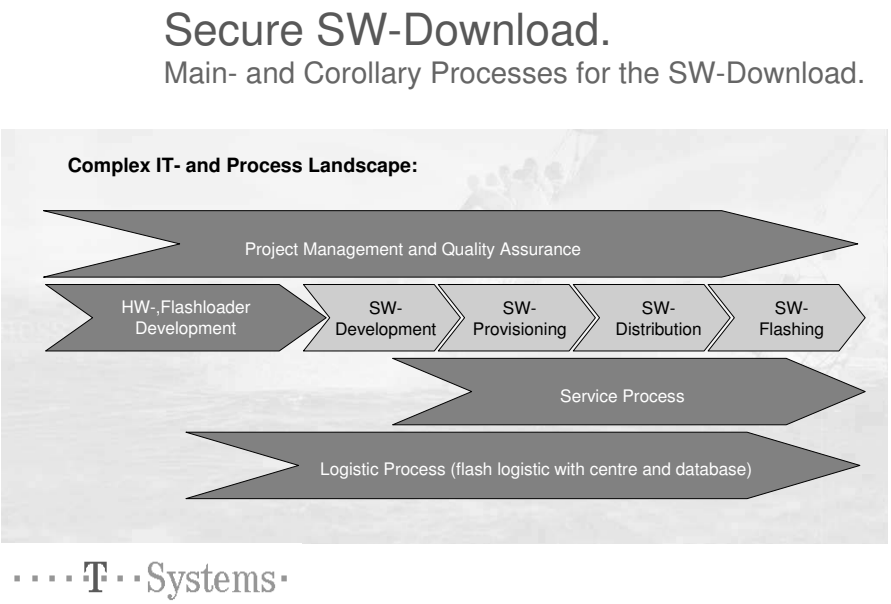

Fig. 1. The software download process

In future, generally a direct connection from a central server into the vehicle will exist. Then remote flashing will be possible by using GPRS or UMTS connections. GSM connections do not have the necessary bandwidth.

The installation of the total process represents a very complex challenge. World-wide operation has to be ensured. This means the process must be accessible and running in a safe, reliable and robust way under different operational conditions.[1] Finally the process of flashing must be organized in such a way that there is trust that in a given vehicle the ECU actually receives the software or flashware intended for it. Beyond that, the process of flashing must be integrated into the service processes.

A further challenge is that of arranging a real and reliable process. This means that the data flow in the process is controllable, accepted as legally binding and consequently comprehensible. In addition, access control must exist which ensures that only authorized persons and instances are allowed to execute appropriate activities. The acknowledgement or rejection of demands for guarantee and demands for warranty possibly depend on this proof.

Therefore, the most important task today is the protection of the software from manipulation and the protection of the flashing process against abuse.

[1] IT security is defined very widely here. It encompasses – besides the well-known demands for availability and integrity – also demands for dependability, controllability, legality and liability. For more details on the concept formation see [2].

3 Prevention Against Attacks and Misuse

However, misuse of flashing techniques is possible. Unauthorized persons may have a vested interest in maliciously updating ECUs with their own software. Most attractive to software tuners is obviously the task of power enhancement, more sportive shock absorber calibration, improved brake behavior, etc. Changes to ECUs in the vehicle immobilizer system can be lucrative as well, especially when circumvention of anti-theft protection functionality becomes feasible.

To prevent these actions and other unauthorized manipulation, and also taking warranty aspects and product liability into account, OEMs have to be able to define exactly which software versions should be executable in ECUs. The information security view specifies this requirement in the security objectives of integrity and authenticity: software in ECUs has to be unmodified and authentic. "Authentic" means the software has been approved by the responsible OEM.

Besides integrity and authenticity, the third security objective, confidentiality, has to be realized in securing software in ECUs. Confidentiality is needed to disable re-engineering attacks or protect new algorithms from access by competitors. Vending of software as an upcoming business case also requires copy protection in order to reduce business risks, which implies another security objective.

Firstly, all the protection needs of the components must be determined by the applications, which are realized in these components. The tachograph components that are presented in detail in this book [12] and the components of the TollCollect system, in particular the on-board unit, have high protection needs. These vehicle components should not be manipulable at any time.

Obvious is the necessity of protection for driver-security relevant components. Especially for this function unauthorized modification may result in damage to persons.

However, due to cost limitations the fulfilment of all security objectives is often not possible and sometimes even not meaningful, e.g., the security needs for some applications might be low. Therefore, special security classes for the ECU have been developed. They will be described in the following section.

4 Security Classes

In order to integrate the complex and heterogeneous ECU landscape of vehicles in a comprehensive security concept, different security classes have to be defined. Such a classification scheme is at present under development by the OEM Initiative Software (HIS, Hersteller Initiative Software).

The security classes ensure two dimensional scalability. The first dimension is defined by the ECU's security objectives. This dimension describes the necessity for security. Therefore the security goal (see Fig. 2) has to be

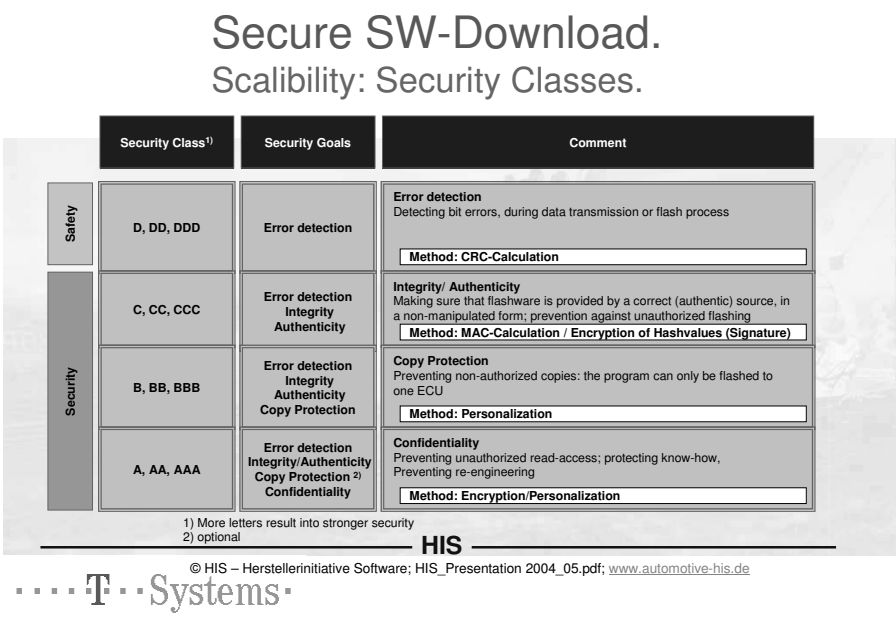

Fig. 2. Security classes by the HIS

defined and the required security strength has to be discussed. As the second dimension, the security capabilities of the ECU have to be identified. The capabilities depend on factors like the processor's performance, given storage space, hardware resistance against attacks, and so on. So the security features finally implemented can be a trade-off of these two dimensions and result in a security class characterized by one to three letters (see Fig. 2).

Some ECUs require less protection than, for example, motor control ECUs which are attractive targets for vehicle tuners (power enhancement) and vehicle thieves (immobilizer system). Thus, there will be ECU software that has to be checked during import against manipulation. It is absolutely irrelevant whether the software comes from the original CD or a copy of the CD. This is right if the OEM is only interested in incorporating non-falsified software into the ECU but not in vending the software.

The minimum requirement for flashing is that the software is correctly loaded into the ECU without any technical error. Technical errors in this context are transfer errors, bit falsifications, bursts etc. Nearly all transmission protocols possess coding techniques for the recognition of such errors. However, the challenge consists of guaranteeing end-to-end protection. This means that, after storage in the flash of the ECU, it must be controllable that the software was genuinely incorporated into the equipment. Most protocols do not guarantee this. Devices which fulfil mechanisms for the realization of this security (safety) goal belong to level D (see Fig. 2). This level contains classes

with the designation D, DD, DDD. The number of letters always refers to the strength of the mechanisms which can be used. The more letters, the higher the security. This applies also to classes that are being introduced.

If an attacker is able to manipulate software, the attacker is also able to change the error-recognizing or error-correcting codes accordingly. Therefore, contrary to popular opinion these codes do not offer any level of protection against active, precise manipulation. Due to this drawback the following class C demands for a protection against manipulation. A protection against manipulation can be realized, for example, by a digital signature or a Message Authentication Code (MAC). A digital signature is based on asymmetric cryptography and an MAC computation applies symmetric cryptography [9, 10, 1].

The goals of the designed classes are the following:

- Level D: Error Detection
- Level C: Error Detection, Integrity and Authenticity
- Level B: Error Detection, Integrity and Authenticity, Copy Protection
- Level A: Error Detection, Integrity and Authenticity, Copy Protection, Confidentiality

In this sense, levels C and D, respectively, ensure basic protection, which guarantees that the developed and approved software in the ECUs is correct. The levels A and B actually protect the business model. Thus, if the software is to be sold then level B will introduce copy protection for the flashware and for level A encryption will be demanded.

A second point is to look at the capabilities and memory availability of ECUs. The processor of an ECU used for infotainment services can store and execute more complex security mechanisms than an ECU, for example, for the roof module. Depending on the characteristics of ECUs a security class at the demanded level can be realized. For example, it is apparent that in level D an error code of 32 bits is more powerful than an error code with 16-bit or 8-bit length. However, the codes require different processor resources and different storage capacity. In level D, the differences are not as significant as in level A, B and C. In contrast to level D, levels A to C require the use of cryptographic functions. However, for the higher levels the required resource can be much larger.

The example below shows data for a signature examination (RSA1024) with today's generally used processors in the automotive industry.

Table 1. Outlay for a signature examination for RSA1024

Processor	Clock	Memory Requirement in kB	Runtime
HC08 (8 bit)	8 MHz	11.5 kB	> 60 s
HC12 (16 bit)	8 MHz	7.5 kB	9.5 s
ST10 (16 bit, non-optimized)	20 MHz	9.5 kB	11 s

Table 1 shows that the capability of the processors varies considerably depending on which safety mechanism with which strength can be used. Thus the capability determines the security standard which can be achieved within the levels. Again, this has to be seen as a trade-off between the security requirements and the capabilities of the ECU.

For example, in level B the question must be asked whether under the given operating conditions an RSA algorithm with 1024 or 2048 key-length or a more efficient algorithm on the basis of elliptical curves should be used. The criterion for the security standard must be the resistance of the assigned cryptographic mechanisms against attacks with respect to today's best technology and presumed advances in the foreseeable future.

The security classes differ basically in their requirements with regard to performance and hardware security of the ECUs. When using digital signatures to secure the integrity and authenticity of software, the public part of the asymmetric key stored in an ECU has to be protected against manipulation. Use of a symmetric algorithm requires protection of the secret key against read-out, which again requires adequate hardware features. Therefore the second dimension is also defined by the hardware security of the ECUs.

Unfortunately, a more detailed description of the security classes according to used mechanisms and algorithms is not possible owing to the absence of publications by the HIS.

5 Security and the Process Chain

The requirements of security classes have to be considered during the design process of ECUs, especially of processors and memories. Embedded systems based on standard processors, which have generally been designed without strict observation of security aspects, will not meet the respective requirements of a security class. Special security processors have been too expensive so far. The so-called Trusted Platform Module (TPM) can be used in the future. This offers good protection for the applicable mechanisms of cryptography and the associated cryptographic keys [11, 7].

A security concept has to take account of the typically long life cycle of vehicles and their components. Changing of keys or algorithms and the fixing of software bugs in security mechanisms is only possible by call-back of vehicles. The security process defined has to be designed in such a way that it remains secure for the total life cycle of vehicles and components. In terms of protection, ECUs are the central and simultaneously weakest elements in the process chain. The security and performance goals will be determined by the functionality and capabilities of the ECU, whereas the security mechanisms to be implemented are affected by the memory available and the long life span of the unit. The issues mentioned are in general technically solvable. They have been solved already for high security requirements, for example, in the defence and banking areas. The challenge for the automotive industry consists

of finding an economical solution. In the current view of OEMs additional costs for secure ECUs are only acceptable in the range of cents.

The development of easy-to-service variants of realization is likewise a challenge. The complexity is shown at present in the roll-out and service for the on-board units (OBUs) of the Toll-Collect system and by the relatively long development times for the digital tachograph.

The realization of protection mechanisms prescribed by the respective security class will not only affect the ECU during the flash process. In achieving continuous protection of software assets leading from the initial release to the final insertion into the ECU in production or service, internal IT systems also have to be integrated into the security concept. Internal systems are used, for example, in software development, quality assurance and release and distribution of software.

6 Security Management Center

The key management center is the central security counterpart of ECUs in vehicles (see Fig. 3). This center generates and administrates the cryptographic keys employed, computes digital signatures and encrypts data. In order to protect keys and their usage against unauthorized access, the key management center has to meet special requirements (physical, organizational and personal).

Finally, key management is essential to the security concept. A variety of different solutions is available. The simplest solution would be to use one key pair per device class. A more complex solution would employ individual cryptographic keys for each ECU. Also, a multi-level PKI (Public Key Infrastructure) solution is feasible.

All solutions as mentioned above have to be taken into account by the OEM. On one hand, such systems secure software assets; on the other hand, they also complicate suppliers' access to ECUs. In the case of warranty or return of devices, suppliers need to update ECUs with special test software for error detection. When access to an ECU is enabled for the respective OEM exclusively, an ECU update with test routines obviously requires assistance by the respective OEM. In order to avoid this additional complication, two separate flash loaders are generally integrated into ECUs. Therefore, the overall security level of any ECU is defined by the flash loader rated with the lowest security level.

Using sophisticated key management techniques as indicated above, OEMs and suppliers can be granted equal access rights to ECUs. Alternatively, an OEM can grant temporary access rights to a specific ECU supplier.

In this context several questions are pertinent: Are the security mechanisms designed to be renewable? Do key changes have to be planned? At which time is that reasonable?

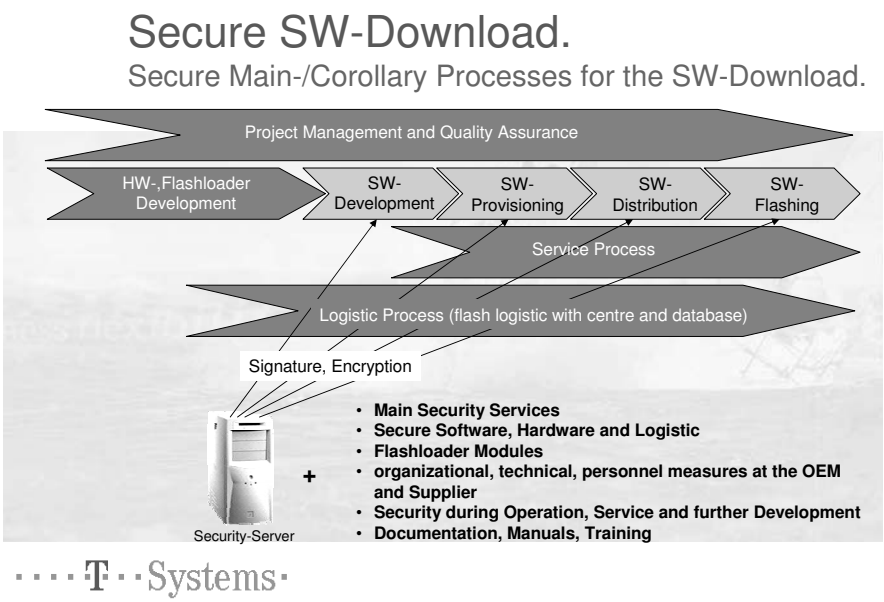

Fig. 3. Integration of the key management center into the download process

There are no clear answers. It is generally sensible to consider that – potentially – an opportunity for manipulation of the mechanisms is opened to an attacker. In order to prevent this, the flashing of security mechanisms and key exchange respectively have to be secured at a higher level than the process of flashing itself. This, again, is only possible if increased demands on hardware security can be imposed on the component.

Further, the question of what times the flashing of the security mechanisms will need has to be discussed. In the following some cases are discussed more concretely. In the case that the security mechanism loaded into the ECU should become compromised for some reason, the flashing of a new version (e.g. containing a new, stronger security mechanism) would be loaded into a potentially insecure environment. In this case a protected area is needed. In the case of compromised keys, it would be desirable to change them as well. But, compromising of the key in the control center is most improbable if a sufficiently high level of environmental security is used in securing the infrastructure. So this question is irrelevant when using asymmetrical security mechanisms for signatures.

A higher probability exists in compromising of symmetric keys due to weaknesses with the protection in a vehicle component. These points of attack are hardly foreseeable and can lie both in the software and in the hardware. Therefore these vulnerabilities can normally be solved only within the further development of the component.

7 Summary

In summary, it may be stated that the protection of flash processes is only possible by implementing all of the security measures described. Only an integrated security concept ensures a secure flash process for vehicle components from development to production and service. This concept has to take both the hardware and software of ECUs into account, as well as adequately reflecting requirements from the system, its infrastructure and the key management center. Today it is one of the most important challenges to coordinate and to integrate all these requirements into a process including hardware development, software development, production, services and after-sales processes.

References

1. Applied Cryptography; Bruce Schneier, John Wiley & Sons, Inc
2. Sicherheit in der Informationstechnik–der Begriff IT-Sicherheit; Rüdiger Dierstein; Informatik Spektrum, August 4, 2004
3. www.automotive-his.de
4. IT-Infrastructure Protection of Telematics Systems Against Manipulation; A. Link. W. Stephan; VDI-Berichte 1728
5. IT-Security in Fahrzeugnetzen; Markus Müller; Eletronik Automotive 4/2004
6. Embedded Security in Automobilanwendungen; Christof Paar; Elektronik Automotive 01/2004
7. Digitale Rechteverwaltung; Marko Wolf, André Weimerskirch, Christof Paar; Elektronik Automotive 2/2005
8. Ganz oder gar nicht; Andreas Link, Markus Müller, Winfried Stephan; Automotive 3.4.2005
9. Fundamentals of Symmetric Cryptography; Sandeep Kumar, Thomas Wollinger; This book.
10. Fundamentals of Asymmetric Cryptography; Thomas Wollinger, Sandeep Kumar; This book.
11. Automotive Digital Rights Management Systems; Marko Wolf, André Weimerskirch, Christof Paar; This book
12. A Review of the Digital Tachograph System; Igor Furgel, Kerstin Lemke; This book.

Secure Software Delivery and Installation in Embedded Systems

André Adelsbach, Ulrich Huber, and Ahmad-Reza Sadeghi

Horst Görtz Institute for IT Security
Ruhr-Universität Bochum
44780 Bochum, Germany
Andre.Adelsbach@nds.rub.de
{huber,sadeghi}@crypto.rub.de

Summary. Increasingly, software (SW) in embedded systems can be updated due to the rising share of flashable electronic control units (ECUs). However, current SW installation procedures are insecure: an adversary can install SW in a given ECU without any sender authentication or compatibility assessment. In addition, SW is installed on an all-or-nothing basis: with the installation, the user acquires full access rights to any functionality. Concepts for solving individual deficiencies of current procedures have been proposed, but no unified solution has been published so far.

In this article we propose a method for secure SW delivery and installation in embedded systems. The automotive industry serves as a case example leading to complex trust relations and illustrates typically involved parties and their demands. Our solution combines several cryptographic techniques. For example, public key broadcast encryption enables secure SW distribution from any provider to all relevant embedded systems. Trusted computing allows to bind the distributed SW to a trustworthy configuration of the embedded system, which then fulfills a variety of security requirements. Finally, we outline the management of flexible access rights to individual functionalities of the installed SW, thus enabling new business models.

Keywords: secure software installation, broadcast encryption, trusted computing, property-based attestation, rights enforcement

1 Introduction

Control unit hardware (HW) and software (SW) in embedded systems used to be tied together as one single product and rarely changed once the system had been shipped. Nowadays, HW and SW in an ECU have become separate products. SW can be updated or upgraded after shipment and add customer

value due to the ubiquitous use of flashable[1] ECUs. Examples are the ECUs in a modern car where updates can increase the engine performance and reduce emission levels. Other examples are upgrades of the car navigation system and updates of the road information data.

Current procedures for installing SW in an embedded ECU are insecure. Details about the deficiencies will be given in Section 2. Historically, these deficiencies didn't matter because SW installation was focused on warranty-based replacement of defective SW. The system owner was informed of costly recalls and received the SW updates free of charge, e.g., when safety-relevant subsystems like airbags or the ESP[2] contained SW bugs. Recently, a paradigm shift has taken place: value-added SW components can be distributed to interested owners and new business models allow the extraction of revenues even after shipment, e.g., when car owners pay annual fees for updates of the navigation system data.

The secure delivery of SW to embedded systems and the management of the corresponding digital rights differs from any existing DRM system known to the authors. First, the distribution currently necessitates a skilled intermediary between SW provider[3] and user because the installation process relies on system-specific equipment which is only available to maintenance personnel. For example, an SW update in a vehicle ECU is usually carried out via a manufacturer-specific diagnostic tester that is only intended for maintenance providers.[4] Second, different classes of such intermediaries exist: depending on their equipment and capabilities, maintenance providers usually have different installation rights. In the automotive example, an uncertified garage might not be granted the right to install SW for safety-relevant ECUs such as the airbag ECU.

Third, a newly developed SW component is not necessarily compatible with any target ECU and the SW of all other ECUs in the embedded system. For example, an average compact-class vehicle contains 40 ECUs, while high-end and luxury class vehicles can have up to 70 ECUs.[5] Secure SW in-

[1] A flashable ECU is a microcontroller capable of reprogramming its memory for application programs and data based on so-called flash memory technology [4].

[2] The Electronic Stability Program (ESP) helps to control a vehicle when it approaches the limits of stability.

[3] By SW provider we mean any party that develops SW for the embedded system, e.g., the original manufacturer of the system and his suppliers, but also independent SW developers.

[4] In the automotive example, maintenance providers such as dealers, garages and road service teams typically carry out the SW installation procedure as the car owner lacks the necessary equipment and skill set [11]. Although diagnostic testers are reported to have been cloned or stolen in some cases, the vast majority of SW updates is still carried out by maintenance providers.

[5] The Volkswagen Phaeton has 61 ECUs [12]. In addition, each OEM usually has different car models with differing ECU configurations. The ECU configuration of a particular model changes during the production life cycle due to an update

stallation must therefore fulfill a variety of requirements regarding security and usability. Last, new business models for embedded systems will induce new requirements. Due to the high value of the embedded system and the potential consequences of system failure, non-repudiation will be an important requirement. For example, if an honest car owner has an accident due to defective SW, his dealer and the SW provider may not be able to deny the installation.

We propose a procedure for secure SW delivery and installation in embedded systems. We combine a variety of different cryptographic techniques to build such a secure procedure. The main contribution of our proposal is the secure installation procedure itself based on Public Key Broadcast Encryption (PKBE) and Trusted Computing. Another contribution will be a requirement model for all parties that participate in a typical distribution and installation setting. To the authors' knowledge, neither a suitable procedure nor a general requirement model has been previously published although several individual requirements have been proposed [3, 11, 28].

The use of the PKBE scheme proposed in [6] has several advantages in this particular setting.[6] First, it enables *efficient* one-way communication from SW providers to a potentially large, but select set of embedded systems, even though they have to be considered *stateless* receivers.[7] Specifically, the length of the message header does not grow with the number of intended receivers[8] as in the case of a standard Public Key Infrastructure (PKI).[9] Second, the proposed PKBE scheme allows the revocation of an unbounded number of receivers. Even if a large number of receivers has been compromised or is to be excluded, messages can still be broadcast to the remaining receivers. Last, it gives non-discriminatory access to the broadcast channel. The public key property allows any (not necessarily trusted) party to broadcast to any chosen set of receivers. Specifically, the manufacturer of the embedded system cannot exclude any SW provider from the broadcast channel or otherwise prevent competition.[10]

---

of HW or SW components [12, 25]. The compatibility of an SW component does not only depend on the target ECU hardware, but also on other ECUs in the vehicle [2, 14, 20].

[6] Broadcast encryption was first introduced in [8] based on private keys. Several improvements were proposed, e.g., in [18]. We refer to public key broadcast encryption in [6].

[7] Stateless receivers contain a fixed set of secret keys which can't be updated after shipment.

[8] Intended receivers are all embedded systems to which the SW provider wishes to distribute a specific SW.

[9] If standard PKI was used on a broadcast channel, the message header length would be $O(|\mathcal{U}|)$, where $\mathcal{U}$ is the set of intended receivers. In the PKBE scheme from [6], the message header length is only $O(r)$, where r is the number of revoked or excluded receivers.

[10] Non-discrimination is also important for maintenance providers at the receiving end: the European Commission Regulation 1400/2002 prevents discrimination of

Trusted Computing is the enabling technology for an embedded system to become a trusted receiver of broadcast messages. Based on minimum additional hardware and cryptographic techniques such as attestation and sealing, an embedded system can be trusted to be in a particular configuration. The assessment of the compatibility of a particular SW component with the embedded system can be based on this configuration. In order to avoid discrimination of certain SW providers, we suggest the use of property-based attestation[11] as introduced in [24].

Section 2 briefly summarizes the work of other authors and illustrates deficiencies of the current SW installation practice. Our overall system model is explained in Section 3. The security requirement model follows in Section 4, while the proposed solution is discussed in Section 5. We conclude and highlight open questions in Section 6.

2 Related Work

Several types of embedded systems exist and specific literature on each type is available. However, we consider a modern vehicle to be the most challenging example, namely due to the specific qualities of SW distribution and installation as outlined in Section 1. In particular, the high number of ECUs and their variants leads to a complex assessment of compatibility. Therefore, we focus on automotive literature and add an example from the field of IT security.

A typical procedure for installing SW in an automotive ECU is described in [4]. It is performed by a so-called flashloader, a standard SW environment that allows for in-system re-programming of ECUs. After initialization of the installation mode, the flashloader erases the programmable memory of the ECU. Then it writes the new SW into the programmable memory. Finally, the procedure ends with the deinitialization of the installation mode.

Current installation procedures rarely apply any cryptographic techniques [4, 5, 14]. The use of signatures has been proposed, but not yet implemented [11, 14, 17]. However, the only signature mentioned in the proposals is that of the manufacturer.[12] If the manufacturer must sign every SW component prior to installation, he is capable of discriminating individual SW providers. In addition, we illustrate several other deficiencies with some examples. First, the intellectual property contained in the SW is not protected, thus allowing reverse-engineering attacks. Second, the installation rights of the maintenance providers are not verified in the course of an installation. Hence anyone with the necessary equipment – including an adversary – can install any (potentially malicious) SW component.

independent maintenance providers. The OEM must give them access to necessary material and technical information, e.g., spare parts and diagnostic equipment.

[11] A similar method called "semantic attestation" has been proposed [9, 10].

[12] The proposals generally do not specify whether they refer to the manufacturer of the embedded system or that of the relevant ECU.

Third, the owner cannot prove that he has legally[13] acquired an SW component that has been installed in his embedded system. Even if the manufacturer applies a signature, the owner can still be accused of having acquired the SW illegally, e.g., without payment of license fees. Fourth, compatibility is not checked by the embedded system. Even if signatures are used, they only prove the source of the SW, not compatibility. SW might be erroneously accepted by an incompatible embedded system due to the manufacturer's signature. Last, no rights management is currently applied. Techniques such as expiry dates or usage counters are not yet implemented and prevent the introduction of more flexible business models. In the automotive example, those techniques would allow to sell additional horsepower or country-specific navigation data for a limited time frame or number of usages.

A framework for international automotive SW installation standards is introduced in [5]. However, it doesn't consider any DRM or security aspects. An infrastructure for installing SW from any external interface is proposed in [11]. Although compatibility is ensured by checking if the hash values of all involved SW components form a valid SW release,[14] no further security aspects are covered. Requirements such as confidentiality, integrity, non-rejection and authenticity are mentioned, but not considered in the proposed architecture and left open to the specific implementation of each vehicle manufacturer. Several other research papers introduce the concept of distributing SW to embedded systems in the field [3, 28], but even if security requirements are mentioned, no specific proposal to fulfill them is mentioned.

A proposal for "end-to-end security" of SW installation in vehicles is made in [17]. However, the signing of the SW component by "an authorized party" is the only protective measure, which provides only a partial solution[15] to the requirements that we will introduce in Section 4. Another proposal for secure SW installation is made in [14]: it contains an authentication phase, in which the diagnostic tester is authenticated, as well as an installation routine, which verifies checksums or signatures of the SW provider. Again, only some of the requirements are fulfilled.[16]

IT security literature often focuses on enforcement of access control policies for downloaded executable content by a secure operating system (OS) or by a secure SW environment which encapsulates the content [15]. However,

[13] By legal acquisition we mean the installation of a compatible SW component from a maintenance provider with the necessary installation rights including payment of all license fees to the SW provider.

[14] An SW release is an SW configuration which has been released by the vehicle manufacturer. An SW configuration in [11] is defined as a valid and operational set of SW components and corresponding coding parameters which can be programmed in the ECUs of a vehicle.

[15] For example, it does not prevent discrimination of independent SW providers as the vehicle manufacturer is assumed to take over the role of the authorized party.

[16] For example, signatures on the receiving end are omitted. Therefore the proposal does not prevent repudiation of a successful installation by the vehicle owner.

embedded ECUs often do not have a standardized operating system. When SW is installed, the whole program memory can be erased and replaced with a new SW component. Therefore, we cannot assume a secure OS or SW environment in *every* ECU; thus the content needs to be analyzed *prior* to installation.

3 Model

3.1 Roles and Objects

The following roles (see Fig. 1) will be used throughout the remainder of this article:

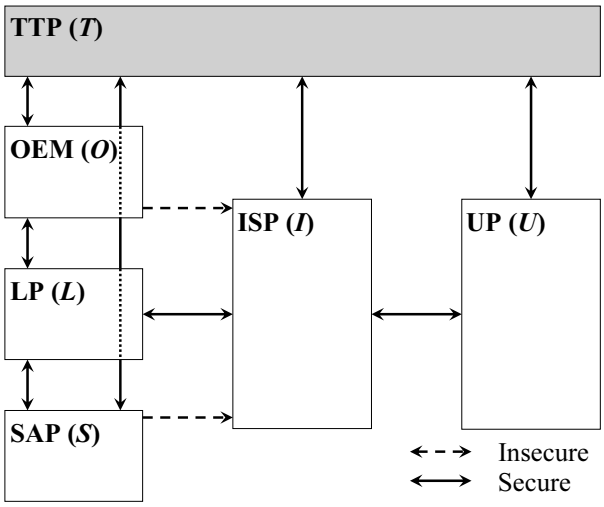

Fig. 1. Roles within the overall model

OEM (O): The Overall Equipment Manufacturer (OEM) develops, assembles and delivers the embedded system to the users. In order to do so, O cooperates with suppliers that develop and/or manufacture components for the embedded system. The initial SW components at shipment time may be either from O or from O's suppliers. Automotive examples are car manufacturers such as Daimler Chrysler, Ford, GM or Toyota.

SAP (S): SW Application Programmers (SAPs) develop SW components for the embedded system. They may either be (i) suppliers that participate in developing and/or assembling the embedded system or (ii) independent application programmers that develop SW components (updates and/or

upgrades) and distribute them after shipment. Automotive examples are suppliers such as Bosch, Delphi, Denso, Siemens and Visteon.
Henceforth, we use the term *SW provider* as a synonym for "OEM or any SAP".

ISP (I): The Installation Service Providers (ISP) maintain the embedded system, i.e., mechanical parts, ECU HW and SW. As part of their maintenance services, they install updates and/or upgrades of SW components. They have equipment that is necessary for the installation procedure and capabilities that allow them to correctly install SW components. Automotive examples are car dealers, garages and road service teams.
The installation rights of I are modeled as clearance levels. Each SW component requires a minimum clearance level $\mathsf{Clear}_{\mathsf{min}}$. I can have any clearance level in $\{1, 2, \ldots, m\}$. If I has clearance level i, it may install any SW with $\mathsf{Clear}_{\mathsf{min}} \leq i$. The highest level m permits the installation of any SW. Without clearance level, no SW may be installed.[17]

LP (L): The License Provider (LP) distributes licenses for SW components that the SW providers O and S have developed. Prior to distribution of a license, L needs to establish terms and conditions with the SW providers in which the model for sharing license revenues is detailed.[18] To the authors' knowledge, automotive examples don't exist yet, but might be established as spin-offs of OEMs and SAPs.

UP (U): The User Platform (UP) is manufactured by O and purchased by the user. The user is interested in SW for U and willing to pay for it if it offers a perceivable value-added. We define U's configuration as the collective information on each SW (and implicitly HW) component that is installed in U. The obvious automotive example for U is a car.
We assume U to have the internal structure depicted in Fig. 2: $\{u_0, u_1, \ldots, u_n\}$ are components of U. In the implementation of an embedded system, the u_i correspond to ECUs. u_0 is assumed to be the trusted computing base (see Section 5.3) and provides a central installation and license service. u_0 is the only component capable of distributing new SW to the other components u_i, $1 \leq i \leq n$. Due to cost constraints, we cannot assume the u_i to be high-performance components, i.e., their computational resources are limited, especially related to cryptographic techniques. The SW dis-

[17] Other models for installation rights can easily be integrated into our proposal. For the purpose of this article, clearance levels serve as an example.

[18] The discussion of licensing models, e.g., pay-per-installation or -usage, is beyond the scope of this article.

tribution from u_0 to the u_i is performed over an internal communication network to which all components are connected.[19]

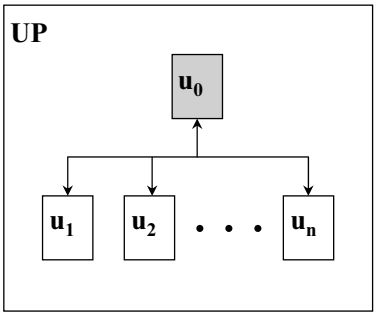

Fig. 2. Internal structure of the user platform

TTP (T): The Trusted Third Party (TTP) has two different certification tasks: first, T creates SW certificates for O and S. These certificates confirm the properties of each newly developed SW component. By *SW properties* we mean characteristic features of SW such as functionality, interfaces, supported protocols, memory and processor requirements, necessary environment, etc. Second, T creates clearance level certificates which certify I's right to install specific SW components. In the automotive example, this role is currently taken over by O. This implies a trust model in which each S must trust O. However, an independent T becomes necessary if O is not fully trusted and discrimination of any S should be avoided. An independent T might evolve out of safety standards authorities such as the NHTSA[20] in the USA or Euro NCAP[21] in Europe.

4 Security Requirements

We consider the security requirements of each party separately. The following terms will be used in this section: when the installation results in *success*, we mean the execution of a complete installation. A *complete installation* includes the installation of a legal SW component and the delivery of a legal license. A *legal SW component* is an SW component whose properties have

[19] If several communication networks coexist, we assume that they are interconnected via gateways and form one coherent network. In the automotive example, this holds true for communication networks – so-called "data busses" – such as CAN, LIN and MOST.

[20] National Highway Traffic Safety Administration, `www.nhtsa.dot.gov`

[21] `www.euroncap.com`

been certified by T and committed by the SW provider. A *legal license* is a license which was legally generated by L and legally acquired by U. By *failure* we mean that no SW is installed, i.e., U's configuration does not change. A *legal I for a specific SW s* is defined to be an I with an authentic clearance level certificate from T which certifies a clearance level sufficient for s. A *legal U* neither requests illegal or incompatible SW nor involves an illegal I.

4.1 OEM Requirements

(OCR) Correctness: The result of the installation procedure must be success if all other involved parties behave according to the specified protocol.

(OPE) Policy Enforcement: O requires enforcement of the following policies:

- **(OPE1) Rights Enforcement.** After acceptance by L, the terms and conditions of O should not be circumvented.
- **(OPE2) Compatibility Enforcement.** An installation will succeed only if the SW and U are compatible. By *compatibility* of an SW and U we mean that the SW properties conform to and are suitable for U's configuration. For example, this implies that the SW must run correctly on U and may not have inconsistent interfaces.
- **(OPE3) ISP Clearance Enforcement.** Only a legal I may install SW.[22]

(OCF) Confidentiality: No party except O and the trusted component u_0 of U may be capable of reading SW developed by O in cleartext *prior to* installation.[23] This is meant to protect the intellectual property contained in O's SW. For example, S may not be capable of simply copying an SW component of O and subsequently distributing it as a proprietary product. However, we only consider conditional access to the SW. Complementary measures, e.g., fingerprinting [13], are beyond the scope of this article.

(OI) Integrity: The installed SW component must be intact and unchanged.

4.2 SAP Requirements

S shares all requirements with O,[24] but has one important additional requirement:

(SND) Non-discrimination: The identity of S may neither influence S's ability to send over the broadcast channel nor the result of the installation procedure. For example, when S_1 and S_2 have each developed a legal SW

[22] For example, this protects O from warranty claims of the user when the user pretends that O and I have colluded to install SW with an illegal clearance level certificate.

[23] This also excludes I from reading the cleartext. However, I will still be necessary in most installation procedures because I has the necessary skill set, installation equipment, maintenance area, spare parts, etc.

[24] Of course "O" needs to be replaced by "S" where necessary.

component with the same properties, S_1 may not be *technically* preferred in the installation procedure.[25]

4.3 ISP Requirements

(ICR) Correctness: This requirement is identical to the requirement OCR.
(INR) Non-repudiation: After each installation procedure, I must be able to prove origin and result of the installation to any honest party.
(ICE) Clearance Enforcement: This requirement is identical to OPE3. For example, this justifies I's effort to obtain a clearance level certificate.
(IND) Non-discrimination: A legal I must be able to install *any* SW component which U requests and which is at or below I's clearance level. For example, the SW provider may not be able to separate ISPs with an identical clearance level into subgroups and exclude individual subgroups from the SW distribution channel.
(IFP) Frame-Proofness: If no installation has occurred, I may not be wrongly accused of treachery, e.g., of having installed SW. We assume the burden of proof is on the accuser.

4.4 License Provider Requirements

(LNR) Non-repudiation: A licensee cannot deny the receipt of a legal license.[26]

4.5 User Requirements

(UCR) Correctness: This requirement is identical to the requirement OCR.
(UNR) Non-repudiation: After the installation procedure, U must be able to prove the result, i.e., either success or failure, to any honest party.
(UIO) Installation Origin: No SW installation may be performed without request by U.
(UA) Authenticity: The installed SW component and the license must be authentic, i.e., as requested by U and sent by the SW provider and L respectively.

[25] Specifically, O may not be able to manipulate U in such a way that U only accepts SW from S_1. However, non-technical influence of O on the user cannot be prevented, e.g., when O advertises for S_1's products.
[26] For example, U cannot receive a legal license and later refuse payment, denying the receipt of a license.

5 Proposed Solution

5.1 Overview

This section provides a summary of the proposed installation procedure (see Fig. 3). The procedure consists of a setup period (Phases A–D) and the actual installation (Steps 1–6). The protocols for these two parts will be detailed in Section 5.2. Section 5.3 introduces the trust and communication channel assumptions on which the protocols rely. Finally, we informally analyze the security aspects of our solution in Section 5.4.

In the setup period, the system parameters, e.g., security parameters of the cryptographic schemes, are chosen. Each I applies for a specific clearance level and is certified by T. This certification is performed once and repeated only if a new I joins the system or existing certificates expire. In parallel, an SW provider who has developed a new SW component submits it to T and requests certification of the SW properties. After certification, the SW provider establishes terms and conditions with L. Both steps need to be done for each new component.[27] Finally, the SW component is distributed to each I via the broadcast channel.

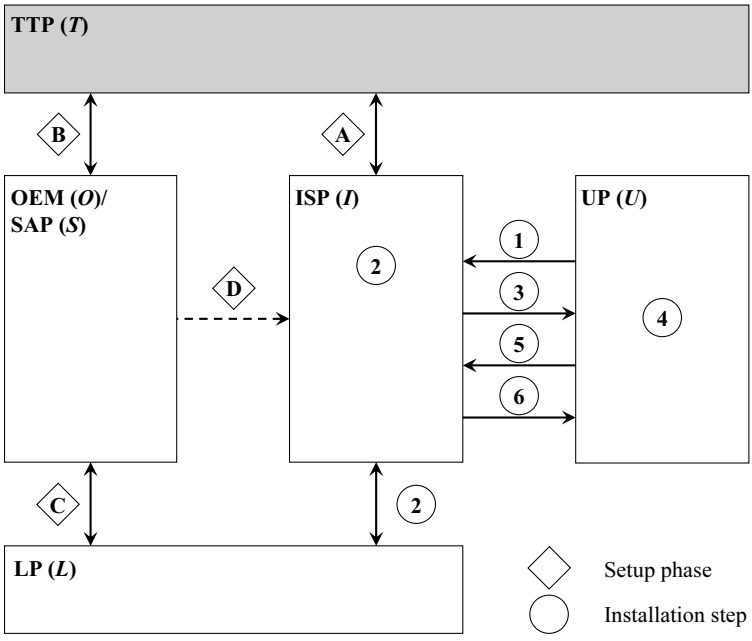

Fig. 3. Installation procedure in six steps (key distribution in Phase B is omitted)

[27] However, an SW provider and L might establish more general terms and conditions which cover a whole set of SW components.

The actual installation starts with an installation request from U to I. I then obtains a license from L. After delivery of SW and license to U, u_0 checks if the SW, the license and I are legal (for definitions see Section 4). If so, u_0 instructs the target component u_i to install the SW. u_0 then confirms the successful installation to I and awaits I's acknowledgement. After receiving the acknowledgement, u_0 instructs u_i to use the SW.

5.2 Protocols

Conventions, Building Blocks and Message Formats

We have tried to minimize the notational overhead of the protocol steps. Although we suppose the notation together with the accompanying text to be self-explanatory, we have summarized all conventions, building blocks and message formats in Appendix A. We refer the reader to this summary whenever related questions arise.

Setup

The setup period consists of four phases A–D (see Fig. 3):

Phase A: Each ISP I applies for certification of a clearance level.[28] The result of the certification process is the clearance level certificate ζ_I which consists of T's signature on the identity and the clearance level of I, generated with T's signing key k_T^{sign}:

$$\zeta_I \leftarrow \mathsf{Sign}(k_T^{\mathsf{sign}}; \mathsf{ID}(I), \mathsf{Clear}(I)) \tag{1}$$

Phase B: T sets up the Public Key Broadcast Encryption (PKBE) scheme, publishes the public keys and provides each U with a set of private keys. In addition, every party[29] generates a private signing key and provides all other parties with the public test key, e.g., using T to distribute the public keys.

Each SW provider[30] *Prov* signs any newly developed SW component s with the signing key k_{Prov}^{sign}: $\sigma_{Prov}^{\mathsf{SW}} \leftarrow \mathsf{Sign}(k_{Prov}^{\mathsf{sign}}; s)$. Then the provider sends $\sigma_{Prov}^{\mathsf{SW}}$ as a property certification request to T, possibly adding the claimed properties and the proposed minimum clearance level. T then verifies the signature $\sigma_{Prov}^{\mathsf{SW}}$, determines the SW properties (or verifies the provider's claim) and assigns the minimum clearance level (or approves the

[28] Many certification models are possible, but we omit their discussion here. One example is a joint definition of clearance level requirements by T, O and S, possibly including spokespersons of I and official authorities.

[29] By "every party", we mean O, L, T and every instance of S, I and U.

[30] That is, either O or S. By *Prov* we mean any of the two roles O and S.

provider's proposal).[31] Finally, T generates the SW property certificate ζ_s, consisting of T's signature on the SW's identity, minimum clearance level and properties. For authentication purposes, T also signs s, resulting in $\sigma_s^{\mathsf{integer}}$:[32]

$$\zeta_s \leftarrow \mathsf{Sign}(k_T^{\mathsf{sign}}; \mathsf{ID}(s), \mathsf{Clear}_{\mathsf{min}}(s), P_1(s), P_2(s), \ldots) \tag{2}$$

$$\sigma_s^{\mathsf{integer}} \leftarrow \mathsf{Sign}(k_T^{\mathsf{sign}}; \mathsf{ID}(s), s) \tag{3}$$

Phase C: During this step, terms and conditions between the SW providers and L are negotiated and committed. Afterwards, L can independently create licenses in (7).

Phase D: The SW provider encrypts T's signature $\sigma_s^{\mathsf{integer}}$ which – by definition of $\mathsf{Sign}()$ – contains the SW component s (4). The signature will allow U to verify the authenticity of s, i.e., the fact that decrypted and certified SW are equal. The broadcast encryption mechanism uses the set of public encryption keys $\mathcal{K}^{\mathsf{enc}}$ to exclusively address the user platforms $\mathcal{U}$. Finally, the SW provider acknowledges the properties with a signature on ζ_s (5) and broadcasts $(s_{\mathsf{enc}}, \sigma_{Prov}^{\mathsf{comm}})$:

$$s_{\mathsf{enc}} \leftarrow \mathsf{Enc}_{\mathsf{PKBE}}(\mathcal{K}^{\mathsf{enc}}, \mathcal{U}; \sigma_s^{\mathsf{integer}}) \tag{4}$$

$$\sigma_{Prov}^{\mathsf{comm}} \leftarrow \mathsf{Sign}(k_{Prov}^{\mathsf{sign}}; \zeta_s) \tag{5}$$

Installation of an SW Component

After this setup phase, the installation procedure for a specific SW component can start:

1. In the first step, U sends a signed installation request σ_U^{req} to I.[33] The request contains the identifier of the requested SW s and the rights $\mathcal{R}_U$ that U desires:

$$\sigma_U^{\mathsf{req}} \leftarrow \mathsf{Sign}(k_U^{\mathsf{sign}}; \rho_U) \quad \text{with} \quad \rho_U := (\mathsf{ID}(U), \mathsf{ID}(s), \mathcal{R}_U) \quad \text{and} \quad \mathcal{R}_U := \{r_1^U, r_2^U, \ldots\} \tag{6}$$

[31] $\mathsf{Clear}_{\mathsf{min}}(s)$ should be based on the safety relevance of s and the required skill set of I. T knows the skill set related to a clearance level based on the clearance level certification in (1).

[32] In a different trust model, O might certify the SW properties. This would significantly reduce the workload on T. However, it would require all S to trust O or result in dispute if O denied fair evaluation.

[33] Several procedures for initiating the installation request can be implemented depending on the trust model. In a classical procedure, the owner of U might physically sign a paper-based installation request and trust I to initiate the installation request on U. In a more technical procedure, the owner might initiate the request himself after identifying himself to u_0 of U with the help of a smart card.

2. I verifies U's signature $\texttt{true} \stackrel{?}{=} \mathsf{Verify}(k_U^{\mathsf{test}}; \sigma_U^{\mathsf{req}})$. If it is valid, I requests a license γ_L for U from L and obtains it in the form (7). The license γ_L is simply a signature of L on U's request. Finally, I signs the installation package $(s_{\mathsf{enc}}, \gamma_L)$ (8):

$$\gamma_L \leftarrow \mathsf{Sign}(k_L^{\mathsf{sign}}; \mathsf{ID}(U), \mathsf{ID}(s), \hat{\mathcal{R}}_U) \tag{7}$$

$$\sigma_I^{\mathsf{inst}} \leftarrow \mathsf{Sign}(k_I^{\mathsf{sign}}; s_{\mathsf{enc}}, \gamma_L) \tag{8}$$

3. I sends the tuple $(\sigma_I^{\mathsf{inst}}, \zeta_I, \zeta_s, \sigma_{Prov}^{\mathsf{comm}})$ to U where ζ_I is I's clearance level certificate, ζ_s the SW property certificate and $\sigma_{Prov}^{\mathsf{comm}}$ the SW provider's commitment to ζ_s.
4. The trusted component u_0 of U finds the appropriate PKBE decryption key and recovers the SW: $(\mathsf{ID}(s), s, \mathsf{Sig}(\sigma_s^{\mathsf{integer}})) \leftarrow \mathsf{Dec}_{\mathsf{PKBE}}(\mathcal{K}_U^{\mathsf{dec}}; s_{\mathsf{enc}})$. Then u_0 verifies the validity of all signatures, certificates and the license: this shows that the SW provider has signed the properties, I has signed the installation package, I has delivered a legal license, I possesses a valid clearance level certificate and the SW property certificate is authentic.
   Afterwards, u_0 verifies that the SW s was indeed requested, the delivered SW is identical to the certified SW (9), I has the necessary clearance level (10), s and U are compatible (11), and the license grants the requested rights:

$$\texttt{true} \stackrel{?}{=} \mathsf{Verify}(k_T^{\mathsf{test}}; (\mathsf{ID}(s), s), \mathsf{Sig}(\sigma_s^{\mathsf{integer}})) \tag{9}$$

$$\mathsf{Clear}(I) \stackrel{?}{\geq} \mathsf{Clear}_{\mathsf{min}}(s) \tag{10}$$

$$\texttt{true} \stackrel{?}{=} \mathsf{Comp}(U; P_1(s), P_2(s), \ldots) \tag{11}$$

   Finally, u_0 determines the target ECU u_i in (12) and re-encrypts s for u_i with a symmetric key k_{u_0,u_i} shared only with u_i (13).[34] Subsequently, u_0 invokes u_i to install the SW component by sending the instruction $instr_{u_i}$ in (14).[35] After the installation, u_i confirms the successful result:

$$u_i \leftarrow \mathsf{Target}(U; P_1(s), P_2(s), \ldots) \quad \text{with} \quad i \in \{1, \ldots, n\} \tag{12}$$

$$s_{\mathsf{enc}}^{u_i} \leftarrow \mathsf{Enc}_{\mathsf{symm}}(k_{u_0,u_i}; s) \tag{13}$$

$$instr_{u_i} \leftarrow \mathsf{Install}(\mathsf{ID}(u_i), \mathsf{ID}(s), s_{\mathsf{enc}}^{u_i}) \tag{14}$$

5. After the installation, U confirms the result of the installation request ρ_U to I. For this purpose, u_0 uses the indicator $ind \in \{\texttt{true}, \texttt{false}\}$ where `true` represents success and `false` represents failure. To ensure consistency, σ_U^{conf} also contains the license γ_L:

[34] The generation of k_{u_0,u_i} will be detailed in Section 5.3. For the time being, we assume the key exists.

[35] Although we omit the equations, we suppose that u_0 and u_i add a Message Authentication Code (MAC) to each of their messages. A MAC is similar to a signature, but uses symmetric keys.

$$\sigma_U^{\mathsf{conf}} \leftarrow \mathsf{Sign}(k_U^{\mathsf{sign}}; \rho_U, \gamma_L, ind) \tag{15}$$

6. I verifies U's confirmation $\mathtt{true} \stackrel{?}{=} \mathsf{Verify}(k_U^{\mathsf{test}}; \sigma_U^{\mathsf{conf}})$ and sends an acknowledgement back to U (16). Within U, u_0 checks the acknowledgement and finally invokes u_i to use the SW component with parameters $(p_1, p_2, \ldots)$ in (17). The parameters tell u_i which functionalities of s it may use, based on the granted rights in the license:

$$\sigma_I^{\mathsf{ack}} \leftarrow \mathsf{Sign}(k_I^{\mathsf{sign}}; \sigma_U^{\mathsf{conf}}) \tag{16}$$

$$\widetilde{instr}_{u_i} \leftarrow \mathsf{Use}(\mathsf{ID}(u_i), \mathsf{ID}(s); p_1, p_2, \ldots) \tag{17}$$

After receiving and verifying the instruction $\widetilde{instr}_{u_i}$, u_i uses the new SW component s with parameters $(p_1, p_2, \ldots)$. u_0 stores all licenses and periodically checks if any of them has expired. When a license expires, u_0 tells u_i to execute the SW with different parameters. For example, the new parameters might instruct u_i to stop using the SW or switch off some functionality, e.g., the additional horsepower in the automotive example. After expiration, u_0 indicates the need for a new license to the user.
If the installation failed, U uses the old platform configuration.

5.3 Assumptions

Although we don't mention expiry dates, nonces and identity checks, we assume they are part of any implementation. Otherwise, replay and impersonation attacks become possible.

Trust Relations

All honest parties are assumed to keep their secrets private, e.g., their signature keys. There are no specific trust assumptions for O, S and I. Assumptions for the other roles are:

L: All SW providers trust L to adhere to their terms and conditions.[36]

U: All parties trust U to (i) keep SW components confidential,[37] (ii) correctly comply with any protocol and (iii) adhere to licenses. However, due to the cost pressure which embedded systems have to experience we cannot assume each component of U to be fully trusted. Therefore, we distinguish between two types of components as introduced in Section 3.1: u_0 is the

[36] This implies a high level of trust in L. It might be reduced to a lower trust level by introducing techniques for tracing L in case of contract violations, e.g., by adding double spender detection. However, due to the focus on secure SW delivery and installation, we omit any advanced licensing techniques.

[37] As addressed in OCF, fingerprinting may additionally deter U from compromising any SW.

Trusted Computing Base (TCB) of U and the u_i, $1 \leq i \leq n$ are only partially trusted.
Every party trusts the TCB u_0 which securely stores U's private keys. In addition, u_0 is U's secure SW installer and internal license server: u_0 only sends SW to a component u_i that possesses a secret key generated by u_i's manufacturer. For example, u_0 may receive this key by means of a certificate from the manufacturer in which the symmetric key is encrypted with u_0's public key: $\mathsf{Sign}(k^{\mathsf{sign}}_{Manu}; \mathsf{Enc}(k^{\mathsf{pub}}_{u_0}; k_{u_0,u_i}))$. u_0 sends SW to u_i only in encrypted form and also authenticates each message to u_i with the shared secret key. Regarding licenses, u_0 stores them and periodically checks if any granted right has expired. If so, u_0 instructs the corresponding u_i to adapt to the new usage rights.[38] We highlight implementation aspects of this TCB, e.g., sealing of the PKBE keys with the properties of a correct u_0, in Appendix C.
The components u_i, $1 \leq i \leq n$, are not fully trusted, but receive appropriate and cost-efficient protection measures. At production time, u_i receives a secret key from its manufacturer. The key is then securely transmitted to u_0 of the embedded system in which u_i is integrated. At production time, the u_i are assumed to be correct, i.e., they don't accept SW from any component but u_0 and they follow u_0's instructions regarding SW installation and execution. Depending on the commercial value of the SW they contain, the u_i receive different protection measures, e.g., tamper-resistant memory[39] for the most valuable components, but only minimal protection for low-value components.[40] Finally, we assume the u_i to be reliable, i.e., complete an installation request in limited time.

T: Every party trusts T. For example, this includes correct certification of SW properties and clearance levels as well as correct publishing of all public keys.

Communication channels

The communication channels are represented in Figs. 1 and 2. All of them are assumed to be *integer*, thus avoiding bit errors. In addition, all but two channels are assumed to be *secure*, i.e., authentic and confidential. The first exception is the one-way broadcast channel, which is neither authentic nor confidential. However, it is *non-discriminatory*: (i) all SW providers can send over the channel and (ii) the channel has global reach, i.e., each I can receive. The second exception is U's internal communication network. Due to cost constraints on U's components, it is only assumed to be integer and reliable. By *reliable* we mean that each message reaches its recipient after a limited

[38] In our protocol, we use SW parameters that u_0 derives for u_i from the license.
[39] For details on tamper resistance see [16] within this book.
[40] For an automotive example, consider the ECUs of the engine, airbags or ESP as valuable components.

amount of time.[41] Finally, the channels between L and I as well as between I and U are assumed to be reliable.

5.4 Security Analysis for Proposed Solution

In order to analyze the security of the protocol steps, we have verified that all of the requirements defined in Section 4 are indeed fulfilled. These verifications are often lengthy and tedious. We therefore omit them here and refer to the full version of this article [1].

6 Conclusion

In this article we have proposed a procedure for secure SW delivery and installation in embedded systems. It integrates installation service providers as intermediaries between SW provider and embedded system and categorizes them into separate clearance levels. Compatibility of SW component and target system is checked prior to installation. The fulfillment of a variety of requirements and the introduction of an elementary license system allows any SW provider to establish new business models that are currently not supported. The SW provider's intellectual property is protected and a variety of digital rights is supported. From the embedded system owner's point of view, the procedure prevents installation of illegal SW and supports warranty claims against the SW provider in case of defective SW with unambiguous evidence.

Public Key Broadcast Encryption (PKBE) enables efficient communication with the embedded systems on an insecure one-way channel. Even if the key material of delivered systems is not changed throughout the lifetime, an unbounded number of embedded systems can be revoked or excluded. The access to the broadcast channel is non-discriminatory, allowing any SW provider to distribute SW components after certification by a Trusted Third Party (TTP). The use of Trusted Computing (TC) concepts induces the necessary trust in the embedded system. Based on minimal TC hardware and a secure operating system kernel, the embedded system can be transformed into a trusted computing base (in an open environment). For example, this allows any SW provider to have trust in the confidentiality of his SW components. The use of property-based SW certification and sealing of the embedded system's configuration replaces the currently criticized attestation mechanisms which might be used for discrimination of individual SW providers.

Several opportunities for future work remain. For example, the need for a TTP should be reduced. One subject of investigation is the generation of the private key material by the embedded systems themselves and subsequent

[41] In a practical implementation, this amount should be in the order of hours or lower even if the channel may occasionally be interrupted.

aggregation into a PKBE infrastructure. In addition, it would be interesting to consider proof-carrying code instead of SW certification by a TTP.

References

1. André Adelsbach, Ulrich Huber, and Ahmad-Reza Sadeghi. Secure software delivery and installation in embedded systems. In Robert H. Deng, editor, *First Information Security Practice and Experience Conference (ISPEC 2005)*, volume 3439 of *Lecture Notes in Computer Science*, pages 255–267, Singapore, Singapore, April 11–14, 2005. Springer.
2. H. Alminger and O. Josefsson. Software handling during the vehicle lifecycle. In VDI Society for Automotive and Traffic Systems Technology [32], pages 1047–1055.
3. BMW Car IT. Das Potenzial von Software im Fahrzeug. Press report, BMW Group, URL `www.bmw-carit.de/pdf/plakate.pdf` – mailto: `info@bmw-carit.de` – file size: 2050 kB, July 22, 2002.
4. Daimler Chrysler AG. Functional specification of a flash driver version 1.3. Specification, Herstellerinitiative Software, URL `www.automotive-his.de/download/HIS%20flash%20driver%20v130.p%df` – mailto: his@mbtech-services.net – file size: 224 kB, June 06, 2002.
5. Christoph Dallmayr and Oliver Schlüter. ECU software development with diagnostics and flash down-loading according to international standards (SAE Technical Paper Series 2004-01-0273). In Society of Automotive Engineers (SAE) [27]. URL `www.sae.org`.
6. Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2003.
7. Euroforum, editor. *Jahrestagung Elektronik-Systeme im Automobil, Fachtag Design – Test – Diagnose elektronischer Systeme*, Munich, February 12, 2004.
8. Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1994.
9. Vivek Haldar, Deepak Chandra, and Michael Franz. Semantic remote attestation – a virtual machine directed approach to trusted computing. *Proceedings of the 3rd Virtual Machine Research and Technology Symposium (May 6–7, 2004, San Jose, CA, USA)*, pages 29–41, 2004. `www.usenix.org/events/vm04/tech/haldar/haldar.pdf`.
10. Vivek Haldar and Michael Franz. Symmetric behavior-based trust: A new paradigm for Internet computing. In NSPW04 [19]. `gandalf.ics.uci.edu/~haldar/pubs/nspw04.pdf`.
11. Cornelia Heinisch and Martin Simons. Loading flashware from external interfaces such as CD-ROM or W-LAN and programming ECUs by an on-board SW-component (SAE Technical Paper Series 2004-01-0678). In Society of Automotive Engineers (SAE) [27]. URL `www.sae.org`.
12. A. Heinrich, K. Müller, J. Fehrling, A. Paggel, and I. Schneider. Version management for transparency and process reliability in the ECU development. In VDI Society for Automotive and Traffic Systems Technology [32], pages 219–230.

13. Jürgen Herre. Content based identification (fingerprinting). In Eberhard Becker, Willms Buhse, Dirk Günnewig, and Niels Rump, editors, *Digital Rights Management: Technological, Economic, Legal and Political Aspects*, volume 2770 of *Lecture Notes in Computer Science*, pages 93–100. Springer, 2003.
14. M. Huber, T. Weber, and T. Miehling. Standard software for in-vehicle flash reprogramming. In VDI Society for Automotive and Traffic Systems Technology [32], pages 1011–1020. URL `www.automotive-his.de/download/presentation-baden-baden-2003-german.zip`.
15. Trent Jaeger, Atul Prakash, Jochen Liedtke, and Nayeem Islam. Flexible control of downloaded executable content. *ACM Transactions on Information and System Security*, 2(2):177–228, 1999.
16. Kerstin Lemke Embedded Security: Physical Protection against Tampering Attacks. This book.
17. Markus Müller. IT-Security in Fahrzeugnetzen. *Elektronik Automotive*, (4):54–59, 2004. ISSN: 1614-0125.
18. Dalit Naor, Moni Naor, and Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.
19. NSPW 2004, editor. *Proceedings of the New Security Paradigms Workshop*, Nova Scotia, Canada, September 20–23, 2004.
20. Uwe Oeftiger. Diagnose und Reparatur elektronisch unterstützter Fahrzeuge. In Euroforum [7].
21. E.I. Organick. *The Multics System: An Examination of its Structure*. MIT Press, Cambridge, Mass., 1972.
22. B. Pfitzmann, J. Riordan, C. Stüble, M. Waidner, and A. Weber. The PERSEUS system architecture. IBM Technical Report RZ 3335 (#93381), IBM Research Division, Zurich Laboratory, 2001.
23. Ahmad-Reza Sadeghi and Christian Stüble. Taming "Trusted Computing" by operating system design. In *Proceedings of the 4th International Workshop on Information Security Applications (WISA'03)*, Cheju Island, Korea, 2003.
24. Ahmad-Reza Sadeghi and Christian Stüble. Property-based attestation for computing platforms: Caring about properties, not mechanisms. In NSPW 2004 [19]. `www.prosec.rub.de/Publications/SadStu2004.pdf`.
25. Martin Schmitt. Software-update, configuration and programming of individual vehicles on the aftermarket with an intelligent data-configurator. In VDI Society for Automotive and Traffic Systems Technology [32], pages 1021–1046.
26. Jonathan S. Shapiro, Jonathan M. Smith, and David J. Farber. EROS: a fast capability system. In *Symposium on Operating Systems Principles (SOSP)*, pages 170–185, 1999.
27. Society of Automotive Engineers (SAE), editor. *SAE World Congress*, Detroit, Michigan, March 8–11, 2004. URL `www.sae.org`.
28. S. Stölzl. Software products for vehicles. In VDI Society for Automotive and Traffic Systems Technology [32], pages 1073–1088.
29. Trusted Computing Group (TCG). TCG Software Stack Specification, Version 1.1. Technical specification, URL `www.trustedcomputinggroup.org`, August 2003.
30. Trusted Computing Group (TCG). TPM Main Specification, Version 1.2. Technical specification, URL `www.trustedcomputinggroup.org`, November 2003.
31. Trusted Computing Platform Alliance (TCPA). Main Specification, Version 1.1b. Technical specification, February 2002.

32. VDI Society for Automotive and Traffic Systems Technology, editor. *Electronic Systems for Vehicles, VDI Berichte 1789, Congress*, Baden-Baden, Germany, September 25–26, 2003. VDI Verlag GmbH Düsseldorf.

A Conventions, Building Blocks and Message Formats

- $\mathsf{ID}()$ is a function that maps a principal or an object to a unique identifier.
- $(\mathsf{GenKey}_{\mathsf{sign}}(), \mathsf{Sign}(), \mathsf{Verify}())$ is a tuple that denotes the key generation, signing and verifying of a digital signature scheme. $\sigma_X \leftarrow \mathsf{Sign}(k_X^{\mathsf{sign}}; M)$ means the signing of the message M with X's signing key k_X^{sign}, resulting in the signature $\sigma_X = (M, \mathsf{Sig}(M))$. $ind \leftarrow \mathsf{Verify}(k_X^{\mathsf{test}}; \sigma_X)$ means the verification of σ_X with the test key k_X^{test}. The result of the verification is the Boolean value $ind \in \{\mathtt{true}, \mathtt{false}\}$.
- $(\mathsf{GenKey}_{\mathsf{PKBE}}(), \mathsf{Reg}(), \mathsf{Enc}_{\mathsf{PKBE}}(), \mathsf{Dec}_{\mathsf{PKBE}}())$ is a tuple that denotes the key generation, user registration, encryption and decryption of a PKBE scheme (see Appendix B). $\mathsf{GenKey}_{\mathsf{PKBE}}()$ is used by T to set up all the parameters of the scheme, e.g., the set of all public keys $\mathcal{K}^{\mathsf{enc}}$ which is available to any party. $\mathsf{Reg}()$ is used by T to compute the set of secret keys $\mathcal{K}_U^{\mathsf{dec}}$ to be delivered to a user U. $\mathsf{Enc}_{\mathsf{PKBE}}(\mathcal{K}^{\mathsf{enc}}, \mathcal{U}; M)$ is used by a (not necessarily trusted) sender to encapsulate a message M with the set of public keys $\mathcal{K}^{\mathsf{enc}}$ in such a way that only the unrevoked users $\mathcal{U}$ can recover it. $\mathsf{Dec}_{\mathsf{PKBE}}(\mathcal{K}_U^{\mathsf{dec}}; C)$ is used by a user U to decrypt C with the private key set $\mathcal{K}_U^{\mathsf{dec}}$ and returns M if and only if the user is unrevoked, i.e., $U \in \mathcal{U}$.[42]
- $(\mathsf{GenKey}_{\mathsf{symm}}(), \mathsf{Enc}_{\mathsf{symm}}(), \mathsf{Dec}_{\mathsf{symm}}())$ is a tuple that denotes a symmetric encryption scheme for key generation, encryption and decryption. The shared key of X and Y is denoted $k_{X,Y}$ (for details on sharing the key, see Section 5.3).
- $\mathsf{MAC}(k_{X,Y}; M)$ is a function that calculates the Message Authentication Code (MAC) of message M under the shared key $k_{X,Y}$ of X and Y.
- $\mathsf{Clear}(I)$ denotes the clearance level of the ISP I. $\mathsf{Clear}_{\mathsf{min}}(s)$ denotes the minimum clearance level that is required for an ISP to install s.
- $\mathsf{Comp}(U; P_1(s), P_2(s), \ldots)$ denotes a compatibility check function that returns `true` if the SW s and U are compatible (see Section 4.1). Otherwise the function returns `false`.
  The compatibility check function $\mathsf{Comp}()$ is computed by u_0 based on the properties $P_i(s)$ of s which have been defined in Section 3.1 on p. 34. For this purpose, u_0 interprets those properties and derives requirements for U such as interfaces, protocols, minimum memory, and minimum processing

[42] In this article, the message is an SW component s. For efficiency reasons, the sender does not encrypt s, but instead encrypts a session key under which s is encrypted. However, to simplify the notation we do not explicitly introduce the session key in our notation.

power. If U fulfills all of these requirements, $\mathsf{Comp}()$ returns `true`, which confirms compatibility of U and s. If any requirement remains unfulfilled, $\mathsf{Comp}()$ returns `false`, indicating incompatibility.

- $\mathsf{Target}(U; P_1(s), P_2(s), \ldots)$ denotes a function which returns the target component u_i, $i \in \{1, \ldots, n\}$ on which the SW s is to be installed.
- $\mathcal{R}_U = \{r_1^U, r_2^U, \ldots\}$ denotes the set of rights that U asks for when sending an installation request. An example for r_i^U is a one-year validity period.
- $instr_{u_i} \leftarrow \mathsf{Install}(\mathsf{ID}(u_i), \mathsf{ID}(s), s_{\mathsf{enc}}^{u_i})$ denotes an order from u_0 to u_i to install $s_{\mathsf{enc}}^{u_i}$.
- $\widetilde{instr}_{u_i} \leftarrow \mathsf{Use}(\mathsf{ID}(u_i), \mathsf{ID}(s); p_1, p_2, \ldots)$ denotes an order from u_0 to u_i to use s with the input parameters $(p_1, p_2, \ldots)$. For example, if $p_i \in \{0, 1\}$ represents a functionality of s, then this functionality is activated for $p_i = 1$ and deactivated for $p_i = 0$.[43] u_0 derives the parameters from the rights $\mathcal{R}_U$ granted in the license.

B Public Key Broadcast Encryption

In a PKBE scheme, any (not necessarily trusted) party can distribute an SW component s on the broadcast channel. Specifically, this holds true for each SW provider S, making the channel non-discriminatory. In the setup phase, T splits the set of all U into a well-chosen subset scheme in such a way that each U is part of several subsets. Two of these schemes were introduced in [18] and extended to the public key setting in [6]. T chooses the security parameters, e.g., key lengths, and generates a public key as well as a private key for each subset. All public keys are supposed to be known to any party while T gives the private key of each subset only to those U that are elements of the subset. T can extract U's keys at any time after setup and gives them to U at production time. For memory efficiency reasons, the scheme minimizes the number of keys that U needs to store.[44] In the automotive case example, the manufacturer of the trusted computing HW might take over the role of T. However, it might also be O if all SW providers trust O.

In the distribution phase, the SW provider first needs to select a set $\mathcal{U}$ of intended users. Then he computes a selection of subsets – called "cover" of $\mathcal{U}$ – in such a way that only the members of $\mathcal{U}$ are contained in the subsets and that the number of subsets remains small.[45] Finally, the provider encrypts s with a session key and, in turn, the session key with the public keys of all subsets in the cover. On the receiving end, each U in the cover has the necessary private keys for decrypting the session key and subsequently s. No other party –

[43] In the automotive example, the functionality might be additional horsepower.

[44] In [6], the authors present two alternatives with user storage requirements $O(\log_2 N)$ and $O((\log_2 N)^2)$ respectively where N is the number of all U.

[45] In [18], the authors present an algorithm that finds a cover of size $O(r)$ where r is the number of revoked users.

specifically, neither the original SW provider nor any U outside the cover – can decrypt the session key, thus providing confidentiality. Although T can decrypt any session key based on the master key, the setup phase can be carried out in such a way that all potential users receive their key set and the master key is destroyed.

In our model, PKBE has one main advantage over a regular Public Key Infrastructure (PKI): PKBE is significantly more efficient regarding message header length, i.e., it needs far fewer encryptions of the session key when a message is sent over the one-way broadcast channel. In addition, PKBE in [6] even comprises a regular PKI. Each user U is contained in a subset of size 1 to which only U holds the private key. Therefore a sender can distribute s even to a very small set of intended recipients by encrypting – in the worst case – the session key with the public subset key of each intended user.

The selection of the intended users might be based on two criteria. Firstly and most importantly, all revoked users are excluded, e.g., when a trusted component u_0 has been compromised and traced. Secondly, all potentially incompatible users might be excluded. For example, if a high-end SW component s can only be installed in a specific luxury class vehicle, the SW provider might exclude any compact class vehicle. However, U still performs a compatibility check. In the example, the compact class vehicle would refuse to install s anyway due to lacking compatibility. Therefore, the second selection step is unnecessary and even increases message header length.

C Implementation

The proposed solution is relevant for an actual implementation. The cryptographic primitives, e.g., signatures, PKBE and symmetric encryption schemes, are readily available and their security has been proven. The roles that we have introduced either exist today or might be taken over by a party that can easily evolve out of existing players in the respective industries.

Trusted computing hardware is currently being developed by several industry groups and standards bodies such as the TCG.[46] An adaptation of the hardware, e.g., Trusted Platform Module or tamper-resistant memory, to an embedded environment seems feasible. In this scenario, the private key material of U is stored in a tamper-resistant memory and all other keys are stored in either tamper-resistant memory or encrypted form. All SW tasks are separated from each other by the operating system, preventing tasks from eavesdropping and modifying the physical memory or processor instructions. Secure operating systems can be based on secure microkernel architectures. For a discussion of these architectures, we refer to [23] which compares, for example, EROS [26], Multics [21] and PERSEUS [22]. Due to the proposed

[46] Trusted Computing Group, `www.trustedcomputinggroup.org`. For details, see [31, 30, 29].

installation procedure, only one component per embedded system needs to be a trusted computing base. This respects cost requirements of the respective industries that prevent the use of trusted hardware in every single component of the system.

Property-based sealing allows to bind the private PKBE keys to a correct configuration of u_0. We derive it from property-based attestation as introduced in [24] (for a similar method see [9, 10]). In contrast to attestation, which only proves U's platform configuration at a certain point of time, sealing allows to permanently bind secret information to a correct platform configuration. For this purpose, a trusted module of u_0 stores the private PKBE keys, but releases them only if u_0 is in a trustworthy configuration defined by u_0's properties. Each time that the task for decrypting PKBE ciphertext calls the trusted module and asks for the private keys, the module determines the current properties of the platform and checks if they match with the properties of a trustworthy configuration. The module releases the private keys only in the case of a match.

If even the task for decrypting PKBE ciphertext is contained in the trusted module, then the SW provider might include an up-to-date set of trustworthy properties in the ciphertext. Subsequently, the trusted module would decipher the trustworthy properties, compare them with the current platform properties and release the session key of the SW only in the case of a match.

Anti-theft Protection: Electronic Immobilizers

Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stüble

Horst Görtz Institute for IT Security
Ruhr University Bochum
44780 Bochum, Germany
{lemke, sadeghi, stueble}@crypto.rub.de

Summary. The automotive industry has been developing electronic immobilizers to reduce the number of car thefts since the mid-1990s. However, there is not much information on the current solutions in the public domain, and the annual number of stolen cars still causes a significant loss. This generates other costs particularly regarding the increased insurance fees each individual has to pay.

In this paper we present a system model that captures a variety of security aspects concerning electronic immobilizers. We consider generic security and functional requirements for constructing secure electronic immobilizers. The main practical problems and limitations are addressed and we give some design guidance as well as possible solutions.

Keywords: electronic immobilizer, transponder, motor control unit, RFID, mafia attack, distance bounding, trusted computing

1 Introduction

Since the mid-1990s, authorities, insurance companies and automotive manufacturers have put much effort into decreasing the number of car thefts in Europe by using electronic immobilizers.[1] An immobilizer system allows the owner of an ignition key to start the car engine. Certainly, improvements have been achieved against car theft through deployment of electronic immobilizers (see, for example, [23, 1, 18]) but also due to better co-operation between authorities in different countries. However, skilled and determined thieves can still overcome electronic immobilizer systems [23], e.g., through applying advanced attacks such as manipulating the control software of the engine just by using the diagnostic interface.[2] Further, [23] addresses several organizational

[1] For instance, Germany is one of the European countries with a high number of stolen cars. This number was 144,057 in 1993 and was reduced to 57,402 in 2002 [23].

[2] For example, around 200 diagnostic devices are currently missing in Germany [9].

weak points: the development and production of electronic immobilizers are not sufficiently secured and the trade of diagnostic devices (including the technical details for electronic immobilizers) cannot be sufficiently controlled due to the annulment of the 'group exemption ordinance'.

As the value loss of stolen cars is large, and this leads to other high costs particularly regarding the additional insurance fees each one of us has to pay, it is worth reconsidering and improving the security of electronic immobilizers.

This contribution starts in Section 2 with a review of the current state of electronic immobilizer systems. Sections 3 to 5 deal with a principal approach to capture the model and to give generic solutions for further development. These parts are extended versions of [21]. Based on cryptographic and security measures this work aims at providing an "open" approach starting from the functional and security requirements on electronic immobilizer systems down to implementation issues. We point out some practical problems, give design rules and discuss some solutions and open issues concerning electronic immobilizers. Finally, in Section 6 we discuss some aspects that are related to electronic immobilizer systems.

2 Electronic Immobilizers: State of the Art

Based on [23] we summarize the previous approaches to counteract car thefts by using immobilizer systems and we sketch some conceptional defects of the current solutions.

An electronic immobilizer system consists basically of two components (see Fig. 1): (i) the transponder that is integrated into the ignition key and (ii) the motor control unit of the car. The transponder proves its identity towards the motor control unit which in turn unblocks the motor engine.

Four generations of electronic immobilizer systems have been built in Germany since 1993. The first generation comprised electrical immobilizers that interrupt the power supply of components such as the starter unless an electronic codeword was entered. This codeword was either integrated in the transponder of the ignition key or entered via a keyboard. Starting from the second generation, the immobilizer function has been implemented in the motor control unit. As long as the immobilizer is active, the motor control unit is blocked. In the second generation fixed codes have been used, but without any encryption on the communication line. In the third generation rolling codes were introduced and encryption was used to secure the communication. Additionally, the components used for the electronic immobilizers are mutually paired, so that an easy replacement of components is prevented. The state-of-the-art electronic immobilizer system belongs to the fourth generation. Here, additional components such as the speed indication are included in the overall authentication process of the immobilizer system.

The number of stolen cars has been reduced by the insertion of electronic immobilizers. Unfortunately, electronic immobilizer systems are still breakable

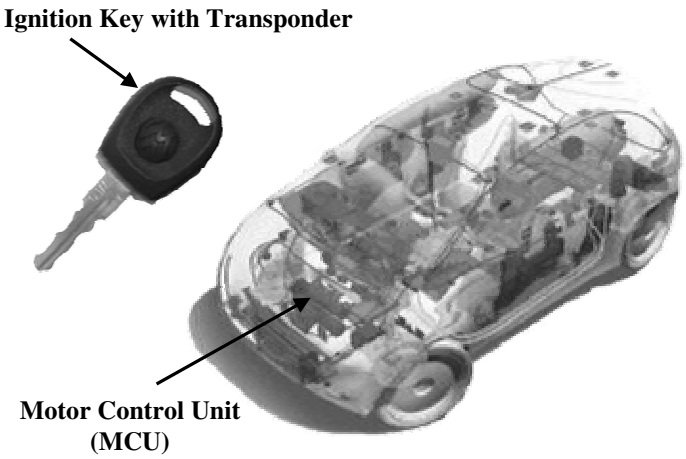

Fig. 1. Main components of an electronic immobilizer

and the number of stolen cars still causes a significant loss. Moreover, there is not much technical information about immobilizers publicly available, and details on the current solutions are rarely known, or only some insights are given.[3]

Commonly car thefts are based on the modification and substitution of components: the cryptographic protocols are not attacked, as yet. Nowadays, car thefts are mainly carried out by criminal organizations and it is assumed that the preparation of substitutes is often done in Eastern European countries [23].

Criminological analyses have revealed that the data structures of persistent memory components such as EEPROM and flash are well known to the thieves. Often only a few bytes of data are changed to bypass the electronic immobilizer system. Using utilities like a programmer tool, the security-sensitive data of motor control units can be read out and modified. It is assessed in [23] that the current security solutions are only engrafted, i.e., security measures are not basically incorporated in the components. In this context it is noted that the implementation details of the components are partly even unavailable to the OEM (Overall Equipment Manufacturer) as the development is outsourced. It is questioned whether information on the concrete information is leaked at the development sites. Further, [23] points out that standardization

[3] For example, by Texas Instruments [18]. Their solution is based on RFID technology and implements a mutual authentication where the underlying cryptographic algorithm used is a proprietary stream cipher. This cryptographic algorithm was completely reverse engineered by [12] at the beginning of 2005. Due to an insufficient key length of only 40 bits brute-force attacks are feasible once the cipher is known.

efforts of the control units lead to common attacks for different vehicle types and save re-engineering efforts in the criminal organizations.

For the optimization of countermeasures, [23] suggests (i) more individual and complex installations of components, (ii) integration of additional components within the electronic immobilizer system, (iii) encrypted storage and encrypted transfer of security-relevant data, and (iv) an enhanced use of combined mechanical and electronical systems.

[23] states that the joint efforts of industry, insurance companies and control authorities should be further improved. Especially, the replacement of security-relevant components should only be feasible using an online connection with the manufacturer's database. Further, it is suggested that the operating license of a stolen or wrecked vehicle is withdrawn. Only after giving evidence on the faultless state of the immobilizer system should a new operating license be issued.

3 System Model

The general model with its components, involved principals, the interfaces between these components and the possible channels to these principals is illustrated in Fig. 2. The principals involved are the vehicle manufacturer M, the car owner O, workshops W (approved by the vehicle manufacturer), control authorities A, insurance companies I as well as trusted third parties.

The electronic immobilizer is embedded in the vehicle's electronics, and consists of three components: The *transponder* T, which is integrated in the *ignition key* of the car, proves its identity towards the *Motor Control Unit* (MCU) that controls the motor engine. The *ignition lock* mainly acts as an interface (e.g., a contactless reader) between transponder and motor control unit, but it can implement some auxiliary functions like a mechanical lock. The communication between the reader and the transponder is radio frequency (RF) based. The transponder obtains its power by the inductive coupling with the RF field that is produced by the reader.

In the following we only briefly consider the involved parties and the trust relations among them. These aspects and the infrastructure required are not the subject of this contribution since our focus concerns the functional and security aspects of electronic immobilizers.

3.1 Trust Relationships

The trust relationships between these parties are very different due to their different interests, and can be very complex. The interests of these parties are manifold: both manufacturers and insurance companies may tolerate a certain threshold on the number of stolen cars. Beyond this threshold, insurance companies may react just by adjusting their loss risk; manufacturers may decide

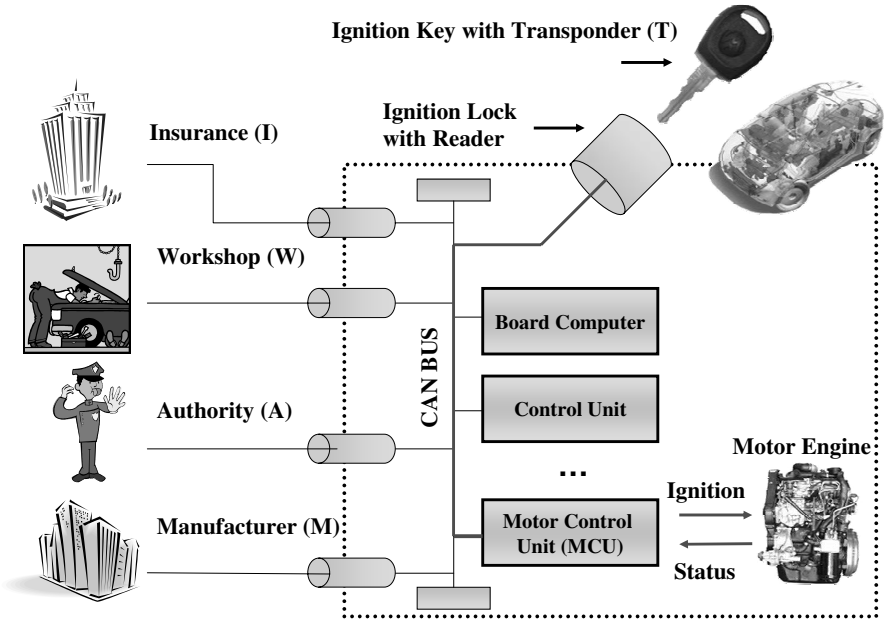

Fig. 2. Infrastructure of the system model

to invest in an improved development of electronic immobilizers to decrease the costs of car insurance or for publicity reasons.

Car owners expect an optimal car theft protection system, including both technical and organizational measures. Car owners and workshops are able to physically access the car components during its operation. One generally assumes that the skills (knowledge and tools) of the workshop employees allow more specific attacks. Manufacturers may be mostly trusted by car owners, while workshops may not. Control authorities are not driven by self-interest and are assumed to play according to the rules to minimize car thefts.

In general, indirect relations (involving a third party) may arise among two principals which lead to more complex relationships. One may consider only two levels of trust, namely full, meaning that a principal is trusted by all other parties, or partial, meaning that this principal is partially trusted by others with respect to certain actions.

The control authority A is the principal who is fully trusted by the other involved principals, but A trusts the other ones only partially. All other trust relations are considered to be only partial.

Primarily, malicious actions are imaginable on behalf of the owner and the workshop. Car thefts can also be made easier because of information leakage at the manufacturer. The car owner is the 'weakest' principal involved who risks being accused both of the modification of components and/or of co-operation with car thieves. The infrastructure and corresponding transactions

(protocols) should therefore guarantee that an honest owner holds evidence for having behaved legally. A further relationship exists between subsequent owners of the car. Also here, the trust relationship is considered to be partial.

As mentioned before, these aspects have an impact on the security of electronic immobilizers; however, they belong to infrastructural and organizational requirements and are not considered further in this contribution.

3.2 Assumptions

Our discussions are based on the following assumptions:

Separation. To keep the system model simple, we assume that the central locking system and the electronic immobilizer are implemented independently, without any interaction, i.e., the corresponding circuits are decoupled.[4]

No Biometrics and no PIN Entry Devices. The ownership of the ignition key is sufficient for authentication. We do not consider biometric measures or PIN entry devices since (i) they require (costly) security devices and (ii) they reduce user-friendliness, e.g., if the car owner wants to lend her car to friends.

Organization. Users are responsible for taking care of the ignition keys as well as the corresponding paper documents. The manufacturers provide a key management infrastructure. We further assume that the identifying data and the secret keys involved are already generated and programmed into the non-volatile memory of both the transponder and the motor control unit (e.g., in a secure production environment).

Physical Access. Towing a vehicle cannot be prevented by any electronic immobilizer!

4 Requirement Analysis

A variety of attacks can be mounted for an unauthorized initiation of the ignition process. Possible threats include cloning or simulating transponders[5] or exploiting any weaknesses in the implementation of security mechanisms or when updating the motor control unit and/or the transponder. Moreover, organizational threats against the transponders and the motor control units are of high importance: critical organizational functions concern, e.g., when users order the replacement of transponders or when new motor control units are to be installed (e.g., in workshops). Fraud can also occur during development and key initialization.

We denote the set of all vehicles by $\mathcal{C}$ and the set of all transponders by $\mathcal{T}$. We call a transponder T *valid*, if there is an approved mapping between

[4] Nevertheless, there are obviously tendencies to integrate both systems [2].

[5] This means being able to construct a device with identical functionality (including the secret initialization data of the target device).

$T \in \mathcal{T}$ and the corresponding vehicle $C \in \mathcal{C}$ where the approval is done by a trusted party (such as the manufacturer) or a trusted component certified by this party.[6]

A simple example is a list signed by the trusted party which contains the identification data (ID) of each transponder ID_T and the ID of the vehicle ID_C to which T is assigned by the underlying mapping. We denote the set of valid transponders by $\mathcal{T}_{valid}$. Informally, the main requirements to be fulfilled by an immobilizer system are:

Correctness: A valid transponder $T \in \mathcal{T}_{valid}$ can always invoke the ignition process of the corresponding car.

Security: For a transponder $T^* \notin \mathcal{T}_{valid}$ it should be infeasible to invoke the ignition process.

To be able to achieve the security objective mentioned above in practice, a variety of technical and organizational building blocks have to be deployed each having its own requirements. In the following we will briefly consider these aspects.[7]

4.1 Security Requirements

In this section we consider the generic security requirements, most of which are wellknown.

Protocol Requirements. Typically, an authentication protocol has to be provided between the transponder and the motor control unit. Security aspects concern protection against active and passive attacks such as eavesdropping the communication between the transponder and the control unit for offline analysis, oracle attacks on the control unit, masquerading and replay attacks, and man-in-the-middle attacks. A type of man-in-the-middle attack is called *mafia fraud* [10], which is of particular concern in the context of wireless systems used for authentication and will be detailed in Section 5.1.

Note that to authenticate the motor control unit, a *mutual authentication* scheme between transponder and motor control unit is reasonable. However, to achieve this in existing vehicles we are faced with the following main problems: Firstly, it is feasible for a skilled adversary to connect a fake MCU to the CAN (Controller Area Network) bus in parallel to the original MCU with the goal to bypass the authentication mechanism later on. To make this hard, the link between MCU and the motor engine has to be separately secured. Secondly, there exists no *trusted path* between the human user and ignition key (e.g., an user interface) yet that can signal to the car owner the result of the authentication (or attestation) protocol.

[6] One may desire procedures that do not require trust in manufacturers. However, in practice manufacturers may not be willing to accept this strategy.

[7] Note that the security requirements should remain fulfilled under different implementations, e.g., when software updates of the motor control unit are done by the manufacturer or if test functions are invoked.

Evaluation. There should be a possibility to verify the correctness of the applied protocols as specified by the immobilizer specification, e.g., by means of emulators checking the communication on the CAN bus. This would increase the trust of users in the underlying immobilizer systems.

Implementation Requirements. Further, technical requirements concern protection against attacks that exploit implementation weaknesses such as inherent leakage (e.g., side-channel attacks [3, 19, 20]), forced leakage (e.g., fault analysis attacks [4, 11]), and vulnerabilities of the logical or physical construction [4].

Organizational Requirements. These security requirements concern life cycle issues, i.e., secure manufacturing, secure initialization (e.g., creation of individual data and cryptographic keys), secure distribution (e.g., transponder maintenance), and secure removal (e.g., destroying of cryptographic keys and components). An important aspect in this context is the requirement that car owners can prove that they are not cheating, e.g., by being able to prove the number of valid transponders even if the car is stolen. Moreover, it should be feasible to detect a complete replacement of the electronic immobilizer system to counteract a typical scenario of vehicle theft where a vehicle is first towed to a garage and subsequently the motor control unit is replaced by another one, which was earlier installed in a junk car.

4.2 Usability and Safety Requirements

Next, we list additional requirements of immobilizer systems starting from presumed functional requirements of the automotive industry, caused by safety and usability reasons:

Time Constraints. The execution time of the authentication must be short. This is obvious since the owner is not willing to wait for the engine to start.

Resource Constraints. The resources (e.g., hardware) are constrained. This is more critical for the transponder.

Maintenance. It should be possible to maintain the transponder on behalf of the owner. This includes cases where the owner wants to block a transponder, e.g., in case it is lost, or add a new one.

Functional Separation. The security functions of the immobilizer should not have an impact on safety aspects, e.g., a successful authentication of the transponder should be valid until the motor is turned off (a running motor should not halt for safety reasons).

No Failure Counters. Failed authentication attempts should not lead to a denial of service.

5 Solutions, Open Issues and Limitations

Based on the requirements of Section 4, we now discuss important aspects to be considered when implementing immobilizer systems.

5.1 Authentication Protocol

Due to the functional requirements on the constrained devices (especially restrictions on the execution time) the use of symmetric cryptography is more efficient than protocols based on asymmetric cryptography.

As mentioned in Section 4.1 the physical link between the motor control unit and the motor engine has to be specially secured. The idea is that an adversary needs more effort to detach the motor control unit. To make the separation hard for the adversary, a possible solution is to weld the MCU to the engine. However, it is also imaginable that the MCU consists of two parts, one part being hard wired with the engine and the other part exchangeable.

For the mutual authentication a *trusted path* between the human user and ignition key (e.g., a user interface) that can signal to the car owner the result of the authentication (or attestation) protocol has to be established. Here, a small light-emitting diode on the ignition key might be a solution. Another solution might be the use of the user interface provided by the on-board computer; however, this implies the assumption that the display cannot be manipulated, which cannot always be guaranteed.

The ISO/IEC 9798-2 three-pass mutual authentication protocol [8] using random challenges is proposed as the basic authentication protocol (see also [13]). The mutual authentication protocol is illustrated in Fig. 3. In the first

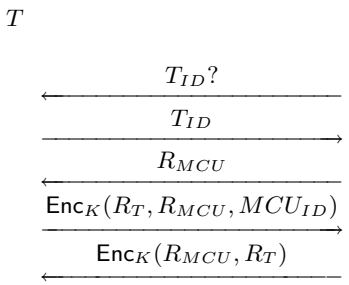

Fig. 3. Three Pass Authentication Protocol ISO/IEC 9798-2

sequence the MCU asks for the identification data T_{ID} of the transponder T. R_{MCU} denotes the challenge raised by MCU. Accordingly, R_T is the random number generated by T. The value MCU_{ID} is assumed to be preprogrammed in the transponder's memory. $\mathsf{Enc}_K(M)$ denotes the encryption of the message M under the secret key K so that both confidentiality and integrity are provided. MCU verifies whether the $\mathsf{Enc}(R_T, R_{MCU}, MCU_{ID})$ is correct. If so, MCU sends $\mathsf{Enc}_K(R_{MCU}, R_T)$ in the last step and MCU invokes the motor engine to start. In case of a failed authentication, MCU returns an error message to T signaling the authentication failure.

Possible cryptographic algorithms for the encryption function include block ciphers, as Triple-DES and AES, and stream ciphers. The cryptographic algorithms Triple-DES and AES are available for direct use, both for encryption and for message authentication codes. A hardware implementation of AES on an RFID based chip is presented in [15].

Preventing Mafia Fraud Attacks. Authentication schemes are used in many applications, but as already observed in [10] mafia fraud attacks cannot be prevented *only* by cryptographic mechanisms. The following scenario demonstrates the mafia fraud: consider a car which is parked next to the house of the car owner and the transponder is located inside the house, e.g., near the entrance. A thief gains mechanical access to the ignition lock and inserts a relaying device instead of the ignition key. The relaying device establishes a radio link which is directed towards the owner's house. Once the transponder is activated by this radio link, the authentication protocol works as specified, which leads to a start of the motor engine.

Here, the adversary does not own the transponder, but the adversary establishes a radio link to the transponder. The adversary makes use of the identity of the transponder, without awareness of the car owner.

Preventing the Activation of the Transponder. Mafia fraud attacks are caused by the RF-based activation of the transponder, which does not require human interaction. Therefore, mafia fraud attacks can be blocked if the transponder cannot be activated by the RF field.

One possible solution is to include an ON/OFF switch on the ignition key which allows the car owner to set the transponder to a non-responsive mode. In the non-responsive mode, the transponder does not answer any requests. Here, the car owner is responsible for ensuring that the transponder is set to a non-responsive mode while not in use. Alternatively, the ignition lock could be used for a mechanical unlocking of the transponder so that the car owner does not need to worry about it.

Distance Bounding Protocol. An upper limit on the distance between two physical entities that are involved in a wireless protocol can be determined by precise timing measurements. Electromagnetic waves propagate with the speed of light c, which is approximately $c = 3 \cdot 10^8$ m/s. The spatial extension Δr of an electromagnetic field after a time Δt is given as $\Delta r = c \cdot \Delta t$. A location which is 3 m away from the origin is reached after 10 ns.

As the transponder and motor control unit exchange two messages, two electromagnetic waves propagate in opposite directions. In real life, additional delays have to be considered for the processing in semiconductor devices: at minimum one clock cycle is passed before the answer can be sent back.

In [14] the authors propose distance bounding protocols. The basic idea is as follows: A series of rapid bit exchanges takes place between the involved parties where the number of bits depends on the security parameter specified. In the corresponding protocol the verifying party V challenges the proving party P, who has access to secret keys, by sending random bits. P has to reply immediately after receiving these bits. The delay time for replies enables V

to compute an upper bound on the distance to P. Some precautions should be taken to guarantee that the responses received by V at the bit exchange originally stem from P, e.g., by a prior commitment by P. A modified distance bounding protocol should counteract mafia fraud attacks also in the case that the random number generator of the transponder is weak.

The suitability of distance bounding strongly depends on high clock rates at the bit exchange sequence. Automotive immobilizers typically work in the carrier frequency range of 100 kHz [16]. Using clock frequencies of 100 kHz, it is not worth implementing the Distance Bounding protocol since the time used for one clock cycle is 10 μs, corresponding to a granularity of distance measurements of 3000 m. At frequencies of 13.56 MHz and above, the embedding of Distance Bounding becomes reasonable, at least for hardware-based solutions at the physical layer, which can minimize the processing delay times.

5.2 Securing the Motor Control Unit

Here, our primary security aim is to prevent the disclosure and modification of secret initialization data of the motor control unit. Further, substitutions of motor control units should be detectable by control authorities later on.

Physical Security. We assume that the core of the motor control unit is a high-performance microcontroller which does not include hardware security mechanisms. In this case, it is recommended to embed the MCU in a tamper-responsive envelope. Note that 'malfunction' of the MCU is a consequence of tampering if the module is encapsulated. An alternative, but still demanding approach, is the development of a secure 'tamper-resistant' high-performance microcontroller. Refer to [4] for an introduction to concepts of physical security.

Since tamper-responsive envelopes are costly, one may be satisfied by using 'only' a small tamper-resistant component that securely stores secrets as long as they are not used. For instance, the *Trusted Platform Module* (TPM) [5, 17] suggested by the trusted computing group[8] (TCG) may be used. The TPM contains a unique certified key, called the endorsement key, that can be used to identify the TPM and thus the motor control unit. Using the TPM, one can also "bind" encrypted content to a specific TPM. This function is called *sealing*, allowing the realization of a secure update function of the control unit software. The *remote attestation* function provided by TPMs allows remote parties to verify the software configuration of the motor control unit using a cryptographically secure hash function. This allows the involved principals to verify the integrity of the installed software of the motor control unit.

Note that an add-on of a TPM requires a secure link between the TPM and each relevant control unit. Further, note that a complete exchange of the TPM and its associated components cannot be prevented either. Nevertheless, the use of a TPM causes higher efforts for the exchange of all associated

[8] `www.trustedcomputinggroup.org`

components. A possible disadvantage might be the complexity that is induced by the TPM.

Interface Security. Attack scenarios as the manipulation of the software with the diagnostic interface are enabled if the design allows to bypass the authentication, either by exploiting flaws or by providing test interfaces which can jeopardize the security of the system. These kinds of attacks can be prevented by careful system design. Software updates need to be verified by the motor control unit (or by the associated TPM) that they were originated by the manufacturer, before the software is modified. An access to test functions should only be granted after a successful mutual authentication, e.g., with the valid transponder.

Auditing. As a complete exchange of the motor control unit (which is, e.g., swapped out from a wreck of the same type) is hard to prevent, mechanisms should be in place which allow the control authorities to determine whether the MCU was originally fitted in the vehicle or not.

A possible solution is an authentication protocol between the control authority and the MCU that transfers one or multiple unique vehicle identification numbers. With this information either the originality is confirmed or the original place of installation can be revealed. However, note that in practice numbers such as the chassis number can still be manipulated.

5.3 Securing the Transponder

The primary aim is to prevent the disclosure and modification of secret initialization data of the transponder.

Transponders include an IC which is optimized for low power constraints. In [18] it was shown that transponders can include EEPROM memory and use it for the long-term storage of initialization data.

It is obvious that transponders should include security mechanisms to counteract both logical and physical attacks. The complexity of the logical functionality of transponders is quite small so that logical protection is definitively manageable; particularly, software updates are typically not foreseen. Regarding physical attacks, the transponders should be equipped with passive protection mechanisms to make tamper attempts sufficiently difficult.

5.4 Replacement of Transponders

The ownership of the ignition key should authorize a principal to start the engine. However, when an ignition key is lost, the owner has to be provided with technical and/or organizational means to block the lost one and obtain and initialize a new one (with new cryptographic keys). There are several solutions imaginable, e.g., those which require a secure channel to the manufacturer or to accredited workshops, and those which do not.

Maintaining Transponders by Infrastructure. In the case of an infrastructure maintained by the manufacturer the car owner is provided with

a new ignition key if the car owner possesses the original paper documents. The initialization of the ignition key can be done by the manufacturer or at authorized workshops. In the latter case, we assume a secure cryptographic link between the transponder and the initialization center.

Maintaining Transponders by Car Owners. Today, it is very costly for car owners to lose a key because only certified workshops can do the replacement. The possibility for car owners to add new transponders and to remove old (e.g. lost) ones independent of the manufacturer would therefore increase both security and usability. In the following discussion, we are assuming that the non-volatile memory of the transponders can be rewritten and that a symmetric key scheme is used.

We propose a solution where the MCU is the central unit that initializes blank transponders (e.g., 'duckling principle' [22]) and that provides appropriate interfaces to authenticate the car owner. As discussed in Section 4, it is essential to ensure that the information on how many valid keys currently exist is counted in a secure way, to ensure that owners cannot deceive insurance companies or buyers of their car. Thus, the MCU cannot be used to store this value, since this information would become unavailable in the case that a car is stolen. Instead, we propose to store the number of valid keys redundantly by all keys. Although this solution requires all transponders to participate in this protocol, it has the benefit that the number of valid transponders can be controlled if at least one valid transponder is available.

To prevent car owners creating secret copies of a transponder, confidentiality of the initialized transponder key has to be guaranteed. One solution is to transmit the cryptographic authentication key in an encrypted form. Therefore, blank transponders have to be shipped with an initial secret key that has to be known by the MCU, requiring some kind of key infrastructure.

Although the proposed solution is, on the one hand, more flexible and improves the privacy of car owners, it requires, on the other hand, more complex handling by the car owner. Moreover, the MCU has to provide an interface to perform the authentication of car owners.

But the most important issue is whether the automotive industry is willing to hand over this maintenance function to car owners, since if a manufacturer-independent maintenance function is available, the manufacturer and the control authority can no longer monitor the personal order of transponders.

5.5 Further Implementation Issues

Random Number Generation. The random number generator should generate an unpredictable sequence of bits (even if the adversary has recorded the previous sequences). A common implementation choice is a pseudo-random generator that is based on a cryptographic cipher and uses two secrets: the key and an initialization value.

Inherent and Forced Leakage. The potential vulnerability of a cryptographic implementation towards inherent and forced leakage cannot be com-

pletely assessed by evaluating the design only. Practical tests should be conducted to examine the susceptibility of the implementation to passive and active side channel attacks. Appropriate defenses for the cryptographic implementation include the use of internal random numbers to de-correlate the inherent leakage of the cryptographic device from the secret data processed. Additionally, a de-synchronization in time is helpful. Fault analysis can typically be averted by an internal verification of the result to avoid the output of faulty cryptograms. For further details refer to [3] and [4] as well as the various contributions of countermeasures against side channel cryptanalysis and fault analysis. Note that an encapsulation as suggested in Section 5.2 makes leakage attacks more difficult, as the microcontroller cannot directly be accessed.

6 Other Directions

6.1 Movement and Positioning Systems

In Section 3.2 it was stated that towing of a vehicle cannot be prevented by an electronic immobilizer. Because of this, adversaries can tow the vehicle to a garage first before they replace components of the vehicle. There already exist sensors (e.g., Hall sensors) which measure the mechanical movement inside the gear of a vehicle. In combination with mobile communication systems (as GSM) alarm events can be signaled to the owner. Additionally, GPS can yield detailed information on the current location of the vehicle. Care should be taken that these sensors cannot easily be detected and disabled or removed before the vehicle is towed.

6.2 Biometrics

These days the use of biometric measures is heavily promoted for identification purposes. Reference [23] discusses biometric measures for use in an electronic immobilizer system. Biometric characteristics that can be used today are

- fingerprints,
- iris scans,
- facial recognition techniques, and
- voice recognition techniques.

Biometric systems are categorized according to their performance metrics. Among them [7] distinguishes

- *failure-to-enrol* or *universality* (the proportion of users who are unable to enrol because of system or human failure),
- *false non-match rate* or *repeatability* (the proportion of cases where a user's biometric measure fails to match that same user's enrolled biometric measures), and

- *false match rate* or *distinctness* (the likelihood that, if a user's biometric measure were to be compared with the enrolled biometric measure of a different person, the two measures would erroneously match).

We do not consider the category *throughput* as it is not relevant for identification purposes in electronic immobilizer systems. Moreover, the *false non-match* and *false match rates* are competing metrics. A decrease of the *false match rate* leads to an increase of the *false non-match* rate, i.e, a legal owner of the car is more frequently refused by the biometric system.

Thönnes and Kruse [23] point out that the biometric devices have to satisfy enhanced environmental requirements in vehicles, e.g., they have to be fully functional within a temperature range of minus 20°C and +70°C. Iris scans and facial recognition systems further require sufficient lighting inside the car, even at night. For cost, reliability and availability reasons, fingerprinting systems are assumed to have the highest potential for biometric identification in cars.

We outline that persons with scuffed or injured fingers may be excluded from using biometric systems based upon fingerprints as they might not succeed to enroll. Such discrimination leads to legal conflicts invoving personal rights. Also safety becomes an important issue if the legal owner is not able to start the car in an emergency because of an injured finger as a result of an accident.

Note further that the biometric device is located inside the vehicle and is subject to modification and substitution as any other component in the vehicle. As the link between the physical measureing device and the control unit of the biometric device is not secure, active modification is feasible in practice.

It is conceivable that car thieves may force the legal owner to initiate enrollment for additional drivers before car theft. An extreme threat is reported in the BBC news [6]: car thieves cut off the finger of the car owner to start the motor engine.

In summary, biometric measures can additionally ensure the identity of the car owner. Nevertheless, in our view they do not justify the additional costs and may cause a great deal of annoyance.

6.3 Carjacking

In European countries carjacking rarely happens [23]. This is different from the countries of America, where it is more widespread. Due to enhanced anti-theft protections there is a certain risk that car thieves will develop other methods to gain access to a car, e.g., by forcing the legal owner to get out.

Technical improvements against an external intrusion are possible and economically justifiable to a certain extent. They should be carefully designed for safety reasons, as it might be necessary to open the car externally in case of an emergency [23]. In addition, control measures are needed to counteract carjacking.

7 Conclusion

We initiated an open approach for designing electronic immobilizers. Herein, we presented and discussed a model, the security and functional requirements as well as solution ideas for constructing secure electronic immobilizers. We pointed out some of the main practical problems and limitations when deploying electronic immobilizers and made some suggestions for implementation. Mainly we considered the aspects of the motor control unit and the transponder which is integrated into the ignition key, but we also propose ideas for the key management by the car owner. A complete physical exchange of an electronic immobilizer system cannot be prevented. However, for the future detection of complete exchanges a cryptographic protocol for control purposes should be foreseen.

References

1. `www.secureyourmotor.gov.uk`.
2. `www.verkehrsunfallforensik.de/pdf/68\_Wegfahrsperren.pdf`.
3. Kai Schramm, Kerstin Lemke, Christof Paar. Embedded Cryptography: Side Channel Attacks. This book.
4. Kerstin Lemke. Embedded Security: Physical Protection against Tampering Attacks. This book.
5. Marko Wolf, André Weimerskirch, Christof Paar. Automotive Digital Rights Management Systems. This book.
6. `http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm`.
7. Public-domain Biometric Applications – Functionality, Performance and Scalability. `www.cesg.gov.uk/site/ast/biometrics/media/perf-and-func-handout.pdf`.
8. *ISO/IEC 9798-2: Information Technology – Security Techniques – Entity Authentication – Part 2: Mechanisms using symmetric encipherment algorithms*. International Organization for Standardization, 1999.
9. Die neue Strategie der Autodiebe. Frankfurter Allgemeine Zeitung, Nr. 40, Seite T1, 2004.
10. Thomas Beth and Yvo Desmedt. Identification Tokens – Or: Solving the Chess Grandmaster Problem. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology – CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 169–176. International Association for Cryptologic Research, Springer-Verlag, Berlin, Germany, 1991.
11. Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *LNCS*, pages 513–525. Springer-Verlag, 1997.
12. Steve Bono, Matthew Green, Adam Stubblefield, Ari Juels, Avi Rubin, and Michael Szydlo. Security Analysis of a Cryptographically-Enabled RFID Device. `www.rfidanalysis.org`, January 2005.
13. Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer, 2003.

14. Stefan Brands and David Chaum. Distance-Bounding Protocols. In T. Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1994.
15. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong Authentication for RFID Systems Using the AES Algorithm. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *LNCS*, pages 357–370. Springer-Verlag, 2004.
16. Klaus Finkenzeller. *RFID-Handbook*. Wiley & Sons LTD, 2003.
17. Trusted Computing Group. TPM main specification. `www.trustedcomputinggroup.org`, Nov 2003. Version 1.2.
18. Ulrich Kaiser. Theft Protection by means of Embedded Encryption in RFID Transponders (Immobilizer). ESCAR conference, Cologne, Germany, November 2003.
19. John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Side Channel Cryptanalysis of Product Ciphers. *Journal of Computer Security*, 8(2/3):141–158, 2000.
20. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.
21. Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stüble. An Open Approach for Designing Secure Electronic Immobilizers. In Robert H. Deng, Feng Bao, HweeHwa Pang, and Jianying Zhou, editors, *ISPEC*, volume 3439 of *Lecture Notes in Computer Science*, pages 230–242. Springer, 2005.
22. Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Security Protocols–7th International Workshop*, volume 1796 of *Lecture Notes in Computer Science*, pages 172–194, Cambridge, United Kingdom, 2000. Springer-Verlag, Berlin Germany.
23. W. Thönnes and S. Kruse. Electronical driving authority – how safe is safe?. VDI Berichte Nr. 1789, 2003.

A Review of the Digital Tachograph System

Igor Furgel[1] and Kerstin Lemke[1,2]

[1] T-Systems GEI GmbH
Solution & Service Center Test Factory & Security
Rabinstr. 8
53111 Bonn, Germany
{igor.furgel, kerstin.lemke}@t-systems.com

[2] Horst Görtz Institute for IT Security
Ruhr-Universität Bochum
44780 Bochum, Germany
lemke@crypto.rub.de

Summary. The European Commission stated the requirements for the digital tachograph system in the regulation No 1360/2002 that has to be fitted into new trucks from 5 August 2005. The digital tachograph system consists of three main components: the motion sensor, the digital tachograph and tachograph smartcards. Each component has to undergo type approval, including an ITSEC/Common Criteria security evaluation. This contribution gives an introduction for the digital tachograph system. Both the technical and non-technical security-related requirements are analysed and (potential) weak points are discussed.

Keywords: digital tachograph, vehicle unit, motion sensor, tachograph cards

1 Introduction

The European Commission regulation No 1360/2002 requires that trucks shall be equipped with a digital tachograph from 5 August 2005. The current analogue tachographs will then be replaced by a digital tachograph system. Note that the EU Directive requires that the digital tachograph is fitted into new vehicles, but not exchanged in vehicles which are already in service.

Originally, the fitting of digital tachographs into all new vehicles was fixed to 5 August 2004. In the meantime, it has turned out that this date is no longer realistic. In a letter dated at 21 April 2004 the EU Commission introduced a moratorium of 12 months starting on 5 August 2004 for the fulfillment of the requirements for the digital tachograph system in all Member States. Now, there are some discrepancies between the European Commission and the EU Parliament concerning the introduction of the digital tachograph (see press

report of the EU Parliament of 13 April 2005, doc. A6-0076/2005). Parliament took the view that all vehicles manufactured after 5 August 2006 should be fitted with this recording equipment. After August 2007, all vehicles put into service for the first time should be fitted with these digital instruments. This question should be agreed in the Conciliation Committee.

Tachographs have been developed to control the working and rest hours of truck drivers as well as the vehicle speed. The EU Commission aims to improve road safety by minimizing accidents that are caused by overtired or speeding truck drivers.

The concern of the EU Directive conflicts with the commercial interest of transport companies. Organisational and technical means have been found to bypass the control of working hours and speed using analogue tachographs. One procedural offense is described in [11]: two drivers swap their vehicles half-way through the working day. If controlled, they show only each second tachograph chart to the control person, so it seems that they stayed overnight at the changing location.

The analogue tachographs do not make use of secured communication channels and could be easily by-passed technically. Ross Anderson showed in [11] that so called "Italian Devices" are available for sale that are fitted in between the analogue tachograph and the motion sensor. This "man in the middle" attack allows control of the forwarding of the number of pulses sent by the motion sensor. There are commercial devices available that leave out 10% or 20% of the pulses on behalf of the driver.

The parties involved in operation are the drivers and transport companies, the workshops, and the enforcing police authorities. Security-relevant manipulations at the tachograph system have to be recognised by the control personnel.

This paper aims to provide an introduction to the binding of the technical components involved as well as the non-technical assumptions on the working environment. Further, some constructional weaknesses are analysed.

2 General Architecture

The general architecture of the digital tachograph system is represented in Fig. 1.

The *tachograph system* consists of the *recording equipment* and *tachograph cards* embedded into the technical and organisational infrastructures (among other key management and fitter workshops) being run by the respective Member State operators.

The *recording equipment* comprises two different elements, as there are the *vehicle unit* (digital tachograph) and the *motion sensor*. It is intended for installation in road vehicles to show, record and store automatically or semi-automatically details of the movement of such vehicles and of certain work periods of their drivers.

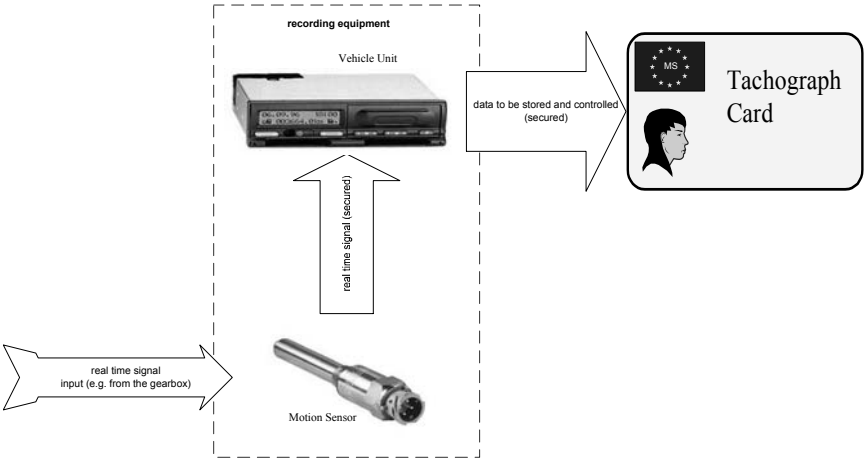

Fig. 1. Architecture of the digital tachograph system

The motion sensor is normally installed in the gearbox of a vehicle and provides a signal representative of vehicle speed and/or distance traveled to the vehicle unit. The latter processes these real-time signals and records the relevant data. The physical information on the vehicle's motion is gained by a mechanical interface.

The tachograph card represents an intelligent storage medium distinguishing different user groups and managing the relevant data. Each user group is equipped with its dedicated tachograph card. The following user groups are defined:

- driver (white card),
- forwarding company (yellow card),
- workshop (red card), and
- control authority (blue card).

After a valid tachograph card has been inserted into one of the two slots for smartcards at the vehicle unit, a mutual authentication between the vehicle unit and the card is performed, so that the vehicle unit "knows" the user operating it. On the other side, the tachograph card is sure that it communicates with a genuine vehicle unit.

3 Functional Specification of the Recording Equipment

The purpose of the recording equipment is to record, store, display, print, and output data related to activities of the system users (i.e. drivers, companies, workshops and controllers). The recording equipment should be fully operational under quite demanding environmental conditions, e.g., the vehicle unit in the temperature range minus 20°C to 70°C and the motion sensor in

the temperature range minus 40°C to 135°C. Data memory content shall be preserved at temperatures down to minus 40°C.

The recording equipment provides the functions for measuring of distances, speed and time. It monitors the activities of the users and creates audit information. A huge amount of data is recorded and stored, e.g., many driver activities must be stored for at least 365 days. Herein, we omit a complete list of data items and refer to the EU Directive [1]. The recording equipment includes different output channels: information can be displayed and printed and there is a secured data download channel. Moreover, functions for the configuration of the recording equipment – the pairing of the vehicle unit with the motion sensor, calibration of the recording equipment and time adjustment – are included. For a complete description of functional requirements we refer the reader to section 3 of [1].

The recording equipment recognizes four modes of operation:

- operational mode,
- company mode,
- calibration mode, and
- control mode.

After a valid tachograph card has been inserted into and recognised (authenticated) by the vehicle unit, the latter switches to a mode of operation according to the rules of [1].

Access conditions on functions provided by and data stored in the vehicle unit depend on its current mode of operation. In order to enforce the prescribed access conditions the vehicle unit implements an integral access control functionality monitoring the current mode of operation and all requests for functions and data. The access control function decides about granting or denying of access to these resources.

4 Functional Specification of the Tachograph Cards

The main purpose of the tachograph cards is to store the relevant data kept by the recording equipment. There are three groups of data to be stored, as shown in Fig. 1.

The electronic part of the tachograph cards is compliant with ISO/IEC 7816 "Identification cards–Integrated circuits with contacts". The tachograph cards should be capable of operating correctly within a five-year period in all the climatic conditions normally encountered in European Community territory and at least in the temperature range minus 25°C to +70°C with occasional peaks of up to +85°C.

The data structures and the access conditions of the files stored on the tachograph cards are specified in section 4 of [1], Appendix 2. This section does not specify the structures used for cryptographic keys and the workshop PIN needed: these structures can be individually defined by each manufacturer.

The minimum storage capacity needed for tachograph data is more than 11 kbytes. On the driver card, this amount is mainly used for the storage of activities, the vehicles used, and the events and faults.

Note that all data files (except for cryptographic keys, which are not specified) can always be read out (without any authentication). The update requires a successful execution of the mutual device authentication and the use of secure messaging. Identification data can never be updated.

Table 1. Data to be stored by the tachograph cards

Data to be stored	**Card type**			
	driver	workshop	control	company
Card identification and security data (initialisation data)				
application identification	x	x	x	x
chip identification	x	x	x	x
IC card identification	x	x	x	x
standard security elements	x	x	x	x
specific security elements	-	x	-	-
Card personalisation data				
card identification	x	x	x	x
card holder identification	x	x	x	x
driving licence information	x	-	-	-
Activity data				
vehicles used data	x	x	-	-
driver activity data	x	x	-	-
daily work periods start and/or end	x	x	-	-
events and faults data	x	x	-	-
control activity data	x	x	x	-
company activity data	-	-	-	x
card session data	x	-	-	-
specific conditions data	x	x	-	-
calibration and time adjustment	-	x	-	-

All data records are organised as ring data structures, so that the newest record will overwrite the oldest record, when the data container is full.

5 Security Requirements

5.1 Recording Equipment

The security of the recording equipment aims to protect

- the data recorded and stored in such a way as to prevent unauthorised access to and manipulation of the data and detecting any such attempts,
- the integrity and authenticity of data exchanged between the motion sensor and the vehicle unit,
- the integrity and authenticity of data exchanged between the recording equipment and the tachograph cards, and
- the integrity and authenticity of data downloaded.

The security requirements for the components of the recording equipment are comparable to the requirements of cryptographic modules, except for the physical security (see Section 8.13).

The general evaluation assurance level defined is ITSEC E3 high or Common Criteria EAL 4+ [2].

5.2 Tachograph Cards

The tachograph card security aims

- to protect the integrity and authenticity of data exchanged between the cards and the recording equipment,
- to protect the integrity and authenticity of data downloaded from the cards,
- to exclude any possibility of falsification of data stored in the cards,
- to detect any attempt and to prevent tampering of that kind.

The Tachograph Card Generic Security Target in Annex 10 of [1] requires that the integrated circuit (IC) of the smartcard is compliant with the

- Smartcard Integrated Circuit Protection Profile – version 2.0 – issue September 1998, registered at French certification body under the number PP/9806 [7], or
- alternatively (see [2]) the BSIPP02: Smartcard IC Platform Protection Profile, 1.0, issued by the "Bundesamt für Sicherheit in der Informationstechnik" [6]

The compliance with these protection profiles is further refined in Appendix 10 of [1].

The general evaluation assurance level defined is ITSEC E3 high or Common Criteria EAL 4+ [2].

5.3 Key Management

The EU legislative for the Digital Tachograph provides in Appendix 11 of the Annex I (B) two different cryptographic systems:

- the *asymmetric* cryptographic system for securing the communication between the vehicle unit and the tachograph card, and
- the *symmetric* cryptographic system with splitting key technology for securing communication between the vehicle unit and motion sensor within the recording equipment.

Asymmetric Cryptography "Vehicle Unit ↔ Tachograph card"

The asymmetric cryptographic system for the digital tachograph is based on the standard Public Key Infrastructure (PKI). The following three hierarchical levels of this PKI are defined:

- European level,
- Member State level, and
- equipment level.

Figure 2 represents the general context of the PKI through the entire hierarchy.

The Digital Tachograph System European Root Policy (Administrative Agreement 17398-00-12 (DG-TREN)) defines the general conditions for the PKI concerned and contains accordingly more detailed information.

The **European Authority** being responsible for the European Root Certification Authority policy is represented by

European Commission
Directorate General for Transport and Energy
Unit E4 – Satellite Navigation System (Galileo); Intelligent Transport
Rue de Mot, 28
B-1040 Bruxelles.

The **European Root Certification Authority** (ERCA) responsible for implementation of the ERCA policy and for the provision of key certification services to the Member States is represented by

Digital Tachograph Root Certification Authority
Traceability and Vulnerability Assessment Unit
European Commission
Joint Research Centre, Ispra Establishment (TP.360)
Via E. Fermi, 1
I-21020 Ispra (VA)

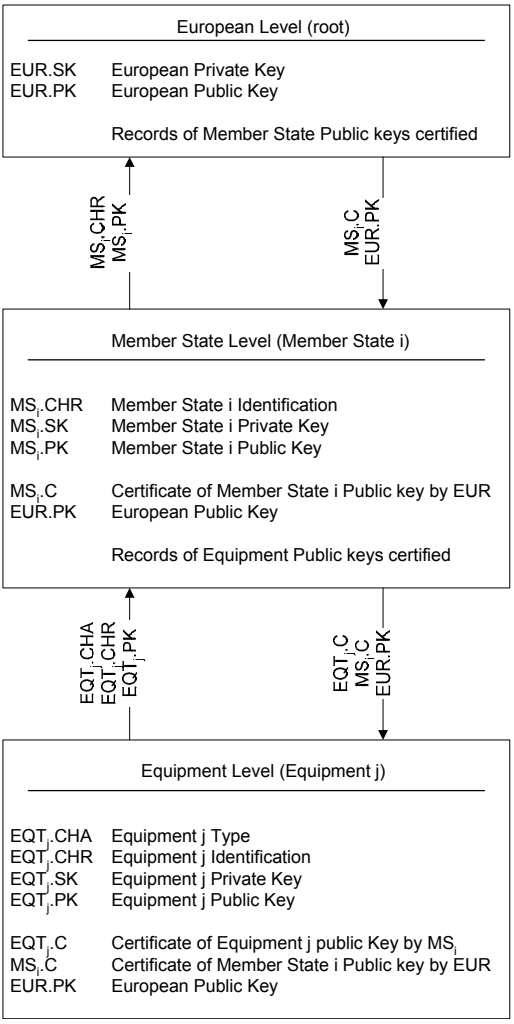

Fig. 2. PKI hierarchy

The ERCA policy [9] is not a part of the Commission Regulation 1360/2002 and represents an important additional contribution. It was approved by the European Authority on 9 July 2004. The ERCA policy is available in electronic form from the web site `dtc.jrc.it`.

At the European level, ERCA generates a single European key pair (EUR.SK and EUR.PK). It uses the European private key to certify the Member States' public keys and keeps the records of all certified keys. A change of the European (root) key pair is not intended.

Each Member State of the European Union establishes its own national **Member State Authority** (MSA) usually represented by a state authority,

e.g. Ministry of Transport. The national MSA runs some services, among others the **Member State Certification Authority** (MSCA). The MSA has to define an appropriate Member State Policy (MSA policy) being compliant with the ERCA policy. At the Member State level, each MSCA generates a Member State key pair (MS_i.SK and MS_i.PK). Member States' public keys are certified by the ERCA (MS_i.C). MSCAs use their Member State private key to certify public keys to be inserted in equipment (vehicle unit or tachograph card) and keep the records of all certified public keys with the identification of the equipment concerned. MSCA is allowed to change its Member State key pair.

At the equipment level, one single key pair (EQT_j.SK and EQT_j.PK) is generated and inserted in each equipment unit (vehicle unit or tachograph card). Equipment public keys are certified by a Member State Certification Authority (EQT_j.C). This key pair is used for

- authentication between vehicle units and tachograph cards,
- enciphering services: transport of session keys between vehicle units and tachograph cards, and
- digital signature of data downloaded from vehicle units or tachograph cards to external media.

The respective MSA (MSA component personalisation service) is responsible for issuing of equipment keys, wherever these keys are generated: by equipment manufacturers, equipment personalisers or MSA itself.

Integrity and authenticity of the entities to be transferred between the different levels of the PKI hierarchy are subject to the ERCA and MSA policies.

The concrete cryptographic algorithm currently being used for the asymmetric cryptographic system is the RSA algorithm. All RSA keys (whatever the hierarchical level) have a length of modulus of 1024 bits.

Symmetric Cryptography "Vehicle Unit ↔ Motion Sensor"

The symmetric cryptographic system for the digital tachograph is based on the splitting key technology. Figure 3 represents the general management of the relevant keys.

The ERCA generates two symmetric partial master keys for the motion sensor: Km_{wc} and Km_{vu}. The first partial key Km_{wc} is intended to be stored in each workshop tachograph card; the second partial key Km_{vu} is inserted into each vehicle unit. The final master key Km results from XOR (exclusive OR) operation between Km_{wc} and Km_{vu}. The additional identification key Kid is calculated as XOR of the master key Km with a constant control vector CV.

The final master key Km and the identification key Kid are used for authentication between the vehicle unit and the motion sensor as well as for an encrypted transfer of the motion sensor individual pairing key Kp from the motion sensor to the vehicle unit. The master key Km and the identification

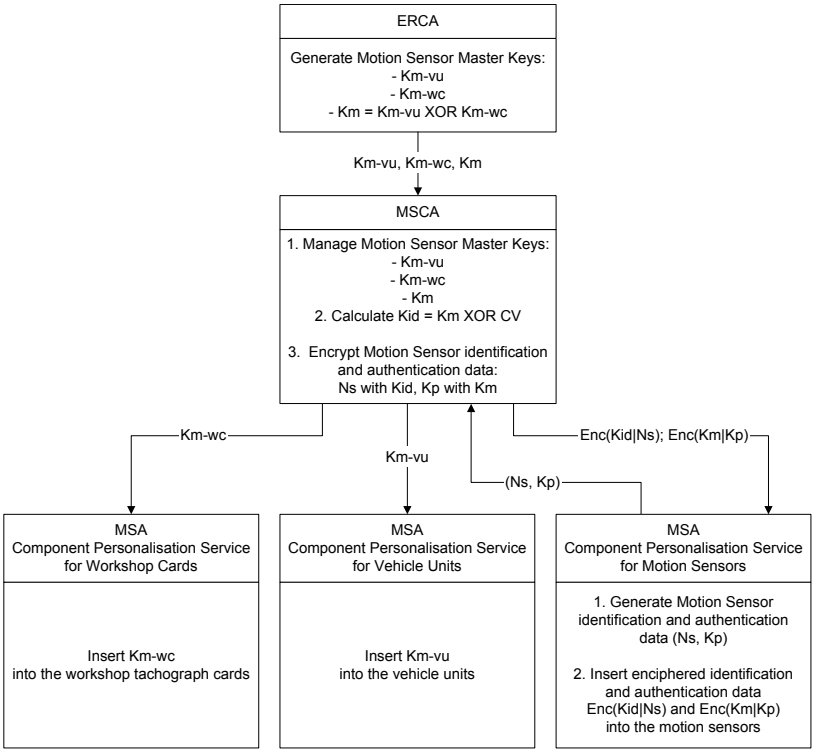

Fig. 3. Key management for the motion sensor

key Kid are used merely during the pairing of a motion sensor with a vehicle unit (see ISO 16844-3 [3] for further details). They are stored neither in the motion sensor nor in the vehicle unit.

Confidentiality, integrity and authenticity of the entities to be transferred between the different levels of the hierarchy within the tachograph system are subject to the ERCA and MSA policies.

The concrete cryptographic algorithm currently being used for the symmetric cryptographic system is the Triple-DES algorithm. Both Triple-DES partial master keys have an effective length of 112 bits (total length of 128 bits).

6 Communication Protocols

6.1 Vehicle Unit and Tachograph Card

Appendix 11 of Annex I (B) of the EU legislative provides two communication phases at the logical level:

- identification and authentication phase and
- operational phase.

During the first phase both communicating parties authenticate each other. As the result of this authentication a common symmetric session key will be established. This session key remains valid till the card is withdrawn from or reset by the vehicle unit. This session key is used for communication between the entities within the operational phase.

Authentication

A mutual authentication between the VU and the tachograph card is required by the EU legislative. Each communicating party should demonstrate to the other that it owns a valid tachograph key pair, the public key of which has been certified by a Member State certification authority, itself being certified by the European certification authority as described above in Section 5.3. The mechanism is triggered at card insertion by the VU. It starts with the exchange of certificates and unwrapping of public keys, and ends with the setting of a session key. Demonstration is made by signing with the equipment private key a random number sent by the other party, which must recover the random number received when verifying this signature and compare the values of the random number sent with the random number received. The relevant protocol is exactly defined in Appendix 11 of Annex I (B) of the EU legislative.

Operation

The operational communication between the VU and the tachograph card can be performed either

1. in plain or
2. using secure messaging in authenticated mode or
3. using secure messaging in encrypted and authenticated mode.

The communication at the logical level succeeds by using the smartcard command set defined in Appendix 2 of Annex I (B) of the EU legislative (see also Section 4). Whether and which secure messaging mode will be used is also defined there.

6.2 Motion Sensor and Vehicle Unit

The EU legislative requires the communication protocol between the motion sensor and the vehicle unit to be compliant with ISO 16844-3 “Motion Sensor interface”. This ISO standard provides two communication phases at the logical level:

- pairing phase and

- operational phase.

During the pairing phase a motion sensor will be "paired" with a vehicle unit. As the result of this pairing a common symmetric session key will be established. This session key remains valid till the next pairing and is used for communication between the entities within the operational phase. Note that this session key is valid for up to 2 years. The pairing can be performed only by an accredited workshop possessing a genuine, valid tachograph workshop card. Generally, the motion sensor implements a set of instructions and plays a passive role, whereas the VU plays an active role in sending these instructions to the motion sensor.

Pairing

Pairing of a motion sensor with a vehicle unit is triggered by a special instruction sent from the VU to the motion sensor. A valid tachograph workshop card must be inserted into and accepted by the VU. After a successful mutual authentication between the workshop card and the vehicle unit, the VU reads out the workshop card part of the master key Km_{wc}. The vehicle unit recomputes the final master key from $\mathrm{Km} = \mathrm{Km}_{vu} \oplus \mathrm{Km}_{wc}$ and the identification key Kid$=$ Km $\oplus$ CV. The vehicle unit authenticates itself by the motion sensor using Kid. A random Triple-DES session key Ks for the operational communication between the VU and the motion sensor is then established. In the last step, the motion sensor authenticates itself by the vehicle unit using the pairing data, the pairing key Kp and Ks. If the mutual authentication was successful, the operational communication continues with the session key Ks. For the concrete details, we refer the reader to [3].

Operation

After having been paired, the motion sensor and the vehicle unit can communicate for operational purposes. Three different kinds of data can be transmitted from the motion sensor to the VU in response to an appropriate instruction:

1. real-time movement pulses,
2. secured value of the pulse counter and
3. secured content of the motion sensor's files being read by the VU.

The real-time movement pulses are continuously transmitted in plain to the VU (when the vehicle is moving) without any security attribute. The frequency of these pulses depends on the instantaneous velocity of the vehicle and the concrete construction of the gearbox, where the motion sensor is mounted (the correct conversion coefficients are determined and stored in the VU during its calibration by an approved workshop).

The motion sensor as well as the connected vehicle unit each runs a pulse counter. Their values are synchronised immediately after the pairing procedure. The VU periodically sends an authentication token to the motion sensor

(at most once per hour), which answers with the random part of the authentication token and the current value of the pulse counter encrypted by the session key Ks. The VU compares

1. the received parts of the authentication token with the respective value having been sent and
2. the current value of its own pulse counter with the value received from the motion sensor.

If these comparisons are successful, the VU "knows" that the motion sensor connected is a correct one and no real-time pulse has been lost or inserted.

In this way the recording equipment assures the correctness of the mean value of the movement data between two subsequent requests for the secured value of the pulse counter. In other words, the trusted input from the motion sensor is supplied as the secured counter value.

Some data (like error messages, serial number, pairing data, etc.) permanently stored in the motion sensor are organised into files, which can be read by the vehicle unit connected. After a special request (including among others an authentication token and the file number) the motion sensor sends the content of the requested file encrypted with the session key Ks.

7 Type Approval of the Components

The prescribed European type approval procedure (see [1], Appendix 9) concerns only three components of the tachograph system – the motion sensor, the vehicle unit and the tachograph card – and comprises four steps:

- Security Certification,
- Functional Certification,
- Interoperability Certification, and
- Type Approval Certification.

7.1 Security Certification

Each of the three components should be certified after ITSEC on the assurance level E3 with the claimed strength of security mechanisms "high". It is also possible to perform the security certification according to the Common Criteria (CC), using a special assurance package E3hAP defined in the "Joint Interpretation Library: Security Evaluation and Certification of Digital Tachographs" [2]. This special assurance package is generally commensurate with the CC Evaluation Assurance Level 4 augmented in the first line by vulnerability analysis for a high attack potential.

[1] also defines a mandatory Generic Security Target for each of three components under consideration, where the required security policies are described. The security policies are defined for the operational life phase of the

tachograph components. The security certificate indicates that the certified product meets the security policy defined in the related security target.

The evaluation and certification processes are usually initiated by the product manufacturer. A prerequisite for issuing a security certificate by an accredited certification body is a successful evaluation having been performed by a licensed evaluation facility.

7.2 Functional Certification

The functional certificate is issued by the national type approval authority. This certificate indicates that at least all functional tests specified by the EU legislative for the tachograph system (Appendix 9 of [1]) have been successfully performed. The functional tests are performed by an accredited laboratory and their results delivered to the national type approval authority issuing the functional certificate. The product manufacturer initiates the functional testing. The functional certificate can normally be gained after issuing the security certificate.

7.3 Interoperability Certification

The interoperability testing aims to ensure that the equipment of different manufacturers works together properly. The product manufacturer requests the interoperability certificate. The application should contain among other things the security and functional certificates. The interoperability certificate is issued by a single central laboratory under the authority and responsibility of the European Commission (JRC Laboratory, Ispra, Italy). This laboratory also carries out the interoperability tests. The interoperability certificate indicates that all interoperability tests specified by the EU legislative for the tachograph system (Appendix 9 of [1]) have been successfully carried out.

7.4 Type Approval Certification

Only a product possessing such a certificate is allowed to be installed into a vehicle. Having received all three certificates – security, functional and interoperability – the national type approval authority issues the type approval certificate for the product in question. The product manufacturer gets a copy of this certificate. The second copy is delivered directly to the central laboratory for interoperability testing (JRC), which updates and publishes on its web site (`dtc.jrc.it`) the current list of products which have achieved the type approval certificate.

8 Conceptual Vulnerabilities

8.1 Long Roll-Out Period

[1] requires that the digital tachograph is fitted into new vehicles, but it does not require exchange of the analogue tachograph built into vehicles already in service. Assuming a truck life-time of more than 20 years, there will be a long period of running both types of tachograph systems in parallel.

As already described in [11] procedural bypasses are possible that allow a company to operate both new and old trucks and to establish a regular change of drivers. If a driver is controlled in a truck, old tachograph charts can be removed in case of a control without notice.

It is recommended to outline a definitive end of the analogue tachographs.

8.2 Delivery and Configuration/Personalisation of Recording Equipment and Tachograph Cards

[1] does not regulate delivery and configuration procedures of the tachograph system components. On the other side, there are special requirements of the criteria for security evaluation (ITSEC and CC) on the properties of delivery and configuration procedures, whereby these could be different for each equipment manufacturer and each Member State. In order to gain harmonised delivery procedures within the whole of Europe, it is necessary to have a common understanding for the delivery methods and responsibilities of the parties participated. Delivery procedures and responsibilities can more easily be understood in the context of the concrete life cycles of the technical components of the tachograph system.

Recording Equipment

Figure 4 visualises the aspects of the delivery and configuration of a VU (the life cycle of the motion sensor is almost the same) in the context of its typical life cycle described in Appendix 10 of [1].

Generation of the VU takes place in the following life cycle phases:

- first initialisation (by the VU manufacturer) and
- initialisation and configuration (by an approved workshop).

During the "first initialisation" phase the cryptographic keys will be loaded into the VU (among other items). The "Component Personaliser Service" of the MSA is responsible for the generation and embedding of cryptographic material into the tachograph equipment (see [9]), wherever such a service is placed. The "Component Personaliser Service" acts upon the National Security Policy of the respective Member State issued by the MSA. So, an appropriate generation as well as a secure delivery of these keys to the VU manufacturer will be assured by the MSA Security Policy.

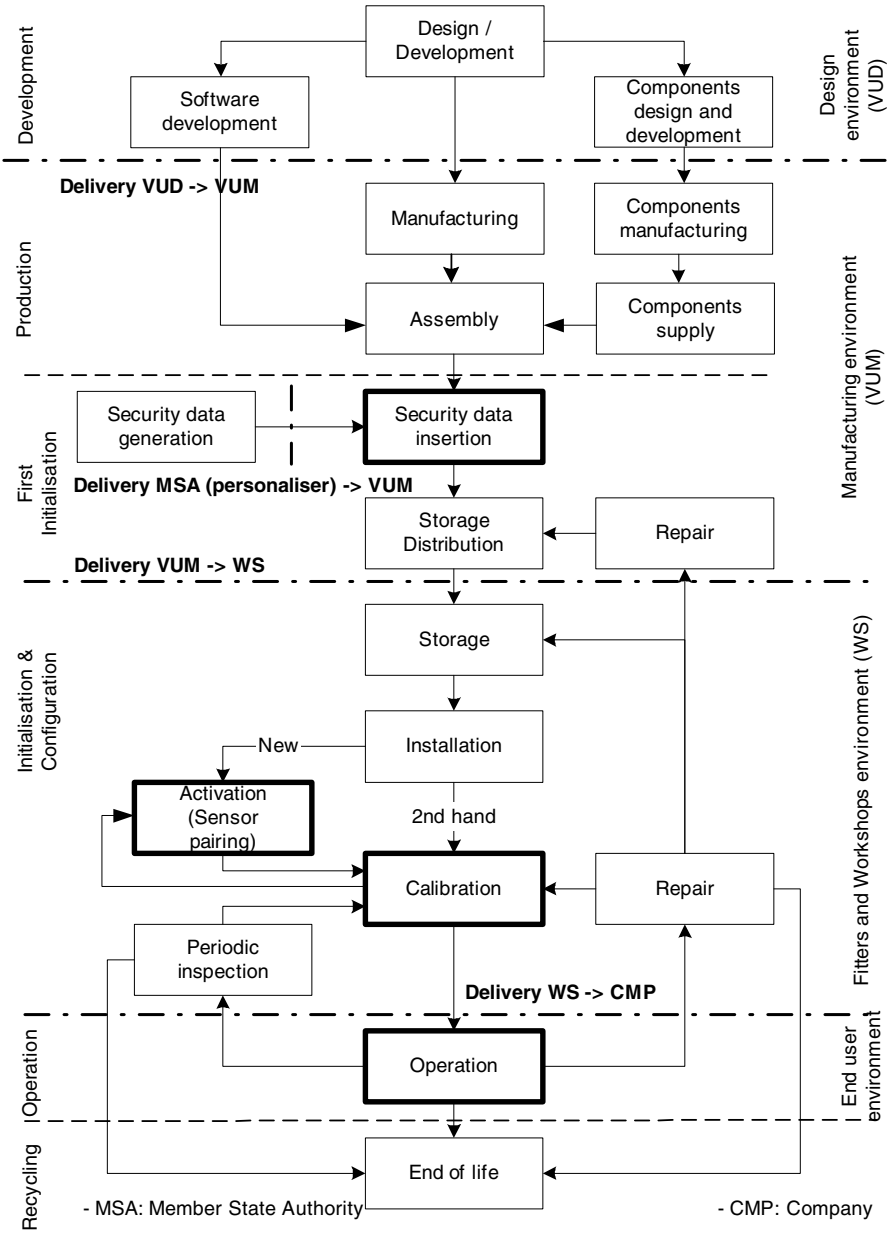

Fig. 4. Vehicle unit life cycle

Configuration of the VU takes place in the phase "initialisation and configuration" by an approved (MSA) workshop. The VU manufacturer should describe the initialisation and configuration procedures supported by the VU

in the User's Manual for the workshop. The technical and organisational environment of the approved workshops should support and provide these procedures. The approved workshops act under surveillance of the MSA.

There are four delivery interfaces (see Fig. 4):

- VU developer → VU manufacturer,
- MSA → VU manufacturer,
- VU manufacturer → approved workshop, and
- approved workshop → company.

Each delivery interface has to be considered in relation to the following questions:

- whether the VU provides any functionality that helps to secure such a delivery interface (this functionality and its usage should then be described in the guidance documents), and
- whether security of the VU depends on the organisational environment during its delivery and what exactly has to be protected by these means. The assumptions about the organisational measures should be described in the guidance documents; compliance with the National Security Policy of the MSA could be helpful.

Tachograph Cards

Figure 5 visualises the aspects of the delivery and configuration of a tachograph card. The phases 1 to 7 correspond to the generic life cycle in [8].

Generation of the tachograph card (TC) takes place in phase 4 "First Initialisation" by and under responsibility of the card manufacturer (CM). At this stage of the card life cycle there is no difference between driver, workshop, company and control cards: the smartcards leave the card manufacturer in the same state.

Configuration of the TC takes place in phases 5 and 6 – Initialisation and Personalisation – by the "Component Personaliser Service" of the MSA, wherever it is placed. The "Component Personaliser Service" acts upon the National Security Policy of the respective Member State issued by the MSA. So, an appropriate generation and loading of card identification, security (among others the key material of high quality) and personalisation data will be assured by the MSA Security Policy. The TC manufacturer should describe the initialisation and personalisation procedures being by the TC in the User's Manual for the MSA Component Personaliser, whose technical and organisational environment will support and provide these procedures. The Component Personaliser acts under surveillance of the MSA.

As one can see in Fig. 5 the smartcards delivered from the card manufacturer to the MSA Component Personaliser do not distinguish between different types of tachograph card. First in the life phase 5 "Initialisation" the card type specific data as Application Identification and Card Certificate will

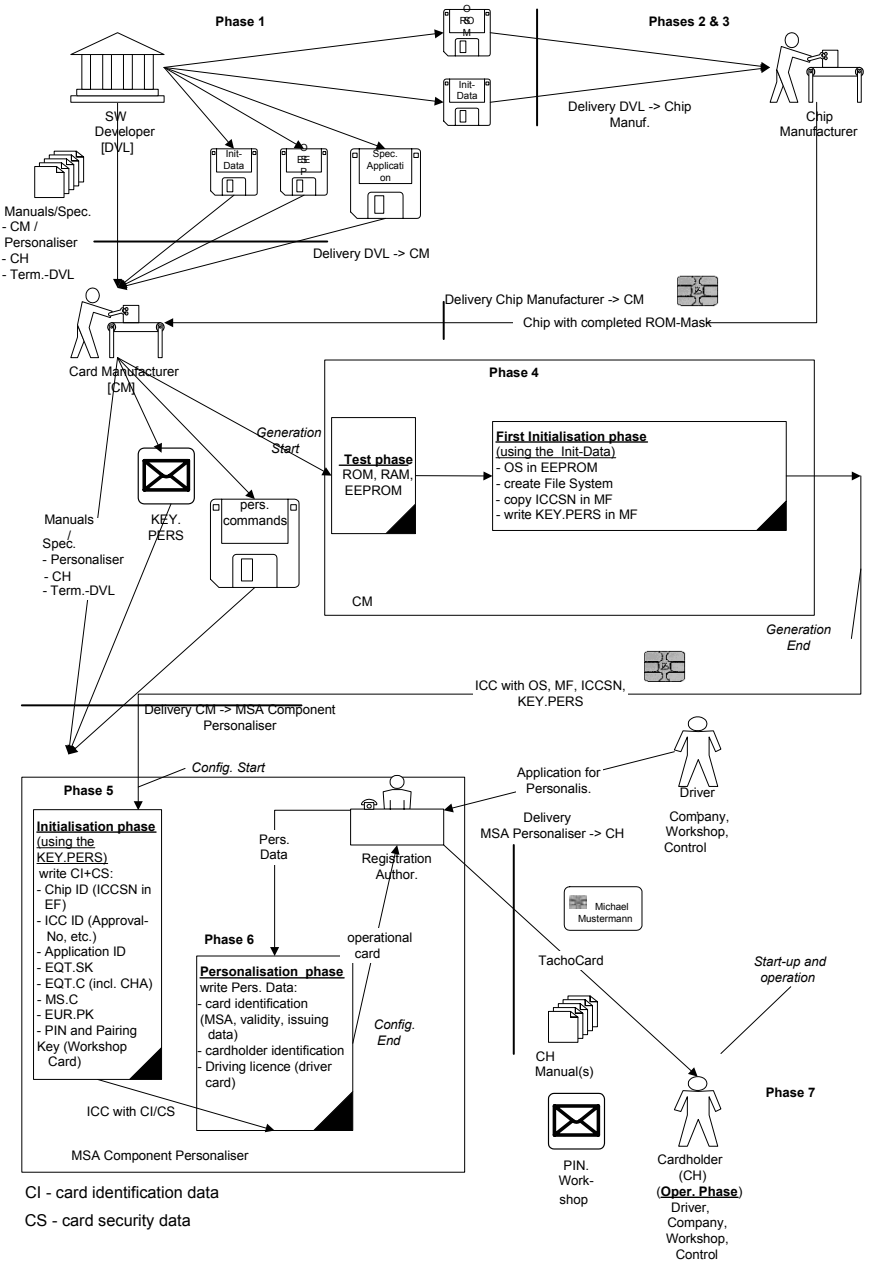

Fig. 5. Tachograph card life cycle

be loaded into the smartcards. After this step the type of tachograph card is unambiguously and irreversible defined: it is either a driver or a company or

a workshop or a control card. Hence, this stage is very appropriate for embedding the ICC into its respective plastic body (so called packaging): white for the driver card, blue for the control, red for the workshop and yellow for the company card.

Generally, the following considerations are helpful: (i) card manufacturer delivers the tachograph cards without any type differences to the MSA, (ii) the MSA first puts the card type specific data into the cards and then (iii) personalisation data concerning the concrete end user of the card (life phase 6 "Personalisation").

There are five delivery interfaces (see Fig. 5):

- developer → chip manufacturer,
- chip manufacturer → card manufacturer,
- developer → card manufacturer,
- card manufacturer → MSA Component Personaliser, and
- MSA Component Personaliser → card holder.

The delivery interfaces have to be considered in the same manner as for the VU (see above).

8.3 Restriction on RSA Key Length

The Public Key Infrastructure which has to be used by the vehicle units and the tachograph cards has already been represented in Section 5.3. The concrete cryptographic algorithm being currently provided for the asymmetric cryptographic system is the RSA algorithm, whereby all RSA keys (whatever the hierarchical level) have a length of modulus 1024 bits (see Appendix 11 of [1]).

In order to gain type approval for a component of the tachograph system this component must obtain security certificate, whereby the strength of security mechanisms must be confirmed to be "high" (see Section 7 and Appendix 10 of [1]). Consequently, also the security mechanisms implementing RSA must be of a high strength, which depends concretely also on the length of modulus.

According to the criteria that may be used for security assessment of the tachograph components (ITSEC and CC), the final decision on the assessment of cryptographic algorithms is made by the security certification body and, eventually, the national competent authority.

It is an established practice to reconsider the cryptographic strength from time to time as new attack techniques are constantly being invented. In order to deal with this fact the responsible national authorities define time restrictions for using cryptographic algorithms as "high-secure" algorithms beforehand. In summary, each Member State has its own guideline for it, which is not made generally public.

In Germany such a guideline is public, and has to be used in the first line in the context of the German Digital Signature Law. Nevertheless, it also

represents a helpful and valid reference for other purposes. The competent national authority is the "Bundesnetzagentur" (Federal Network Agency, `www.bundesnetzagentur.de`), which issues an annual bulletin for the appropriate cryptographic algorithms (the last publication is dated by 2 January 2005).

Concretely, this bulletin of the German authority allows the use of the 1024-bit sized RSA as "high-secure" in the context of the German Digital Signature Law merely until end of 2007 (see Fig. 2).

Table 2. RSA, Length of modulus

		period		
		till end of 2007	till end of 2008	till end of 2009
length of modulus	at least	1024	1280	1536
	recommended	2048	2048	2048

As far as we know the assessments of other authorities lie in a similar range or are even more stringent.

Due to this circumstance the evaluator has to bind his evaluation verdict on the restriction that the assessment of the strength of mechanisms is reconsidered at the latest by end of the period of validity for the current implementation. The product is then reconsidered and eventually recertified.

If at a future time the RSA algorithm with 1024-bit key length cannot be considered as "high-secure", the operators of the tachograph system (the respective MSAs) will face the problem of running the system on equipment (VU and TC) no longer compliant with the requirements of Annex I (B) of [1]. The question of liability in this case is an important one.

8.4 Maintenance of the PKI for the Tachograph System

Proceeding from the issue of length of the RSA modulus touched upon in Section 8.3, the question of maintenance of the PKI arises for the tachograph system.

[1] prescribes a fixed length for the RSA modulus at 1024 bits. Moreover, it does not provide any option for changing the European key pair (EUR.SK, EUR.PK) controlled by the ERCA. The latter plans the life time of this root key pair for a period of 30 years (see [9], section 4.2.6). Of course, ERCA assumes that technological progress over the next 30 years will render its IT systems obsolete. Nevertheless no change procedure for the European key pair has been defined.

According to [1] MSCAs are allowed to perform a regular change of their key pairs (MS_i.SK, MS_i.PK). The European Root Security Policy restricts the use of the MSCA key pair to a period of at most two years starting from

certification by the ERCA (see [9], section 5.3.4). The ERCA will issue a new certificate MS_i.C for each new MSCA key pair. This means that there will be some generations of equipment certificates EQT_j.C having been issued using different MSCA private keys. This does not represent a problem as long as the trustworthiness of the certificate chain can be determined by using the common European public key.

It is the core of the conflict between the prospective necessity to exchange the European key pair against a longer one and the current specification of the tachograph system not providing and not allowing any exchange procedure: neither of a technical nor of a procedural nature.

Also from the logistical point of view, the distribution of a new EUR.PK and of the respective Member State certificates MS_i.C among the single equipment units (already in operation) may be a big challenge for the entire tachograph community. Hence, in order to be prepared for such problems, the tachograph community should already have a modification procedure for the European key pair in place.

8.5 Master Key for the Motion Sensor

A universal master key for the motion sensor is used for the pairing between the digital tachograph and the motion sensor. The key is split (XOR) into two parts. One half is stored in the workshop cards, while the other half is stored in the vehicle unit (see Fig. 3). If this master key is compromised, the security of the overall tachograph system is jeopardised. In detail, the consequences are as follows:

- The observation of the pairing protocol discloses the session key. An additional simulating device can be used in operation that is placed between the vehicle unit and the motion sensor.
- The initiation of the pairing protocol can be invoked without use of a workshop card.
- Cloning of motion sensors is feasible.

There are no precautions taken to replace this master key in the actual key management design.

8.6 PIN Management

The PIN is transferred in clear by the digital tachograph to the workshop card as part of the mutual authentication sequence. Internally, the workshop card verifies the PIN value and returns an “OK” or “KO” message. A “KO” return value results in a failed authentication. After five unsuccessful PIN authentication events, the workshop card is blocked, and cannot be reset to an operational mode.

There are two technical issues. First, the PIN is sent in clear by the tachograph; an interception of the PIN value at the communication line is possible.

Second, the protocol does not include a message authentication for the return value. The return code "OK" can stem from an additional device in between that blocks the "Verify PIN" request.

In Appendix 10 of [1] (functional requirement UIA_302) it is stated that the PIN mechanism is intended "for the vehicle unit to ensure the identity of the card holder, it is not intended to protect workshop card content". This is unusual for smartcard specifications that protect the use of certain functions on human or device authentication.

Typically, workshop cards are used by all employees of a workshop, which results in organisational questions concerning PIN handling. It is probable that the PIN is noted at the workshop card or stored in additional software tools. Further, it might be that the PIN is not entered manually at the keyboard of the digital tachograph, but sent by a standard automotive interface device (e.g. by using the CAN-bus).

Note that the digital tachograph is not a physically secure PIN-entry device. There are no physical requirements to secure the path between the keyboard and the processing unit. Nevertheless, this is a minor issue, formally mended by the organisational measure M.Approved_Workshops (see Section 8.8).

8.7 (Non-)Trustworthy Physical Motion Information

The physical information used to derive the motion data is generated outside the motion sensor. If the physical environment of the motion sensor can be manipulated, the motion information gained can depend on this manipulation. This weak point has been outlined by Ross Anderson [12].

The non-technical requirement M.Mechanical_Interface, which is part of the security target of the motion sensor in Appendix 10 of [1], says "*Means of detecting physical tampering with the mechanical interface must be provided (e.g. seals)*". The environmental conditions within a gearbox include heat and dust. It remains doubtful whether typical measures for tamper evidence such as security seals can be used in this hostile environment reliably.

8.8 (Non-)Trustworthy Workshops

Ross Anderson outlined [12] that about 70% of the employees of workshops have been in conflict with the law in the United Kingdom. The non-technical requirement M.Approved_Workshops, which is part of the security targets for both the motion sensor and the digital tachograph in Appendix 10 of [1] says "*Installation, calibration and repair of recording equipment must be carried by trusted and approved fitters or workshops*". There are serious doubts whether this personal assumption holds in reality.

The non-technical requirement M.Faithful_Calibration ("*Approved fitters and workshops must enter proper vehicle parameters in recording equipment*

during calibration."), which is part of the security target of the digital tachograph, causes similar doubts.

The operator of the tachograph system (the MSA, e.g. the Ministry of Transport) will audit and license the approved workshops. In this way the operator can control the workshops.

8.9 (Non-)Trustworthy Drivers

The security target for the digital tachograph requires that "*Drivers must play by the rules and act responsibly (e.g. use their driver cards, properly select their activity for those that are manually selected, ...)*" (M.Faithful_Drivers).

Drivers are controlled by the tachograph system. Naturally, supervised persons try to find ways to circumvent an external control.

Currently the actual working hours needed can exceed the limits of the EU Directive; conflicts with the company are probable when acting in accordance with the rules of the digital tachograph system.

Another issue might be the financial interest for the driver to enter a false activity (e.g. working time instead of rest time). The monitoring of speeding events is once more an issue that incite manipulation by the driver.

Easier problems may arise because of the handling of the extensive functions of digital tachographs.

8.10 Frequency of Controls

The procedural requirement M.Controls says "*Law enforcement controls must be performed regularly and randomly, and must include security audits*". Due to today's heavy truck traffic (especially in transit) an adequate frequency of controls can probably not be guaranteed.

The future will show how many obviously "out of function" digital tachograph systems will be notified at controls. Note that the company has 15 days before the tachograph has to be exchanged/repaired.

8.11 Training of Control Personnel

The control authority is the only instance that is actually assumed to play by the rules of the tachograph system. It is up to the control personnel to prove any indication of manipulation of the digital tachograph system which would allow the truck to be taken for detailed analysis. Note that the physical security requirement "tamper evidence" requires that trained controllers examine the digital tachograph and the motion sensor in detail (especially, from all sides).

By checking the tachograph's data memory it is assumed that irregularities of other drivers can be detected. It has to be clarified how this is going to be handled.

Miscalibration of the vehicle unit leads to deviations between the motion data measured and the real motion data. The basic parameters for the vehicle characteristics are named "*w, k, l, tyre size, speed limiting device setting, current UTC time, current odometer value*". w and k are constants that give the number of impulses per kilometre, l is the effective circumference of the wheel tyres in millimetre. These parameters are also noted on an installation plaque that is placed in or near the tachograph. Nevertheless, if the workshop does not act by the rules, it may become probably difficult for the control personnel to check the correctness of these vehicle characteristics.

8.12 Controlled Data: Download Data and Paper Printouts

The easiest way for the control personnel is to download the data of the digital tachograph and of the tachograph card of the driver to a laptop. In this way, the authenticity of the downloaded data is guaranteed. Provided that appropriate software is used, an automatic check of irregularities can be carried out, which simplifies the control efforts.

An alternative method is that the control personnel checks the paper printouts.

As printouts can be faked, if the original paper is used there is no assurance that existing printouts were originally generated by a digital tachograph. If the paper printouts were not generated by the digital tachograph of the controlled vehicle, serious doubts remain if the data files of the tachograph card are not checked. Note that the control personnel can produce printouts from the vehicle unit also during a control.

Another disadvantage of paper printouts is the fact that the check for irregularities has to be performed manually, which is a difficult and time-consuming task.

8.13 Physical Security of the Recording Equipment

The Generic Security Targets in Annex 10 of [1] require that the motion sensor and the vehicle unit should fulfil the following requirements being reprinted below:

"*If the motion sensor is designed so that it can be opened, the motion sensor shall detect any case opening, even without external power supply for a minimum of six months. In such a case, the SEF shall generate an audit record of the event (It is acceptable that the audit record is generated and stored after power supply reconnection).*
If the motion sensor is designed so that it cannot be opened, it shall be designed such that physical tampering attempts can be easily detected (e.g. through visual inspection)." (RLB_106)

"*If the VU is designed so that it can be opened, the VU shall detect any case opening, except in calibration mode, even without external power supply for a minimum of six months. In such a case, the SEF shall generate an audit record (It is acceptable that the audit record is generated and stored after power supply reconnection).*
If the VU is designed so that it cannot be opened, it shall be designed such that physical tampering attempts can be easily detected (e.g. through visual inspection)." (RLB_206)

The second implementation choice ("case cannot be opened") calls for tamper-evident measures. The first choice ("case can be opened") is unusual for requirements on the physical security of cryptographic modules. Especially, after the first case opening, it has to be assumed that the motion sensor or VU is no longer trustworthy, as its internals may be modified. Moreover, it is important to add that the workshops are allowed to open the case in calibration mode for maintenance.

Note that tamper evidence calls for a frequent and random control of the vehicle (see Section 8.10) as well as for a careful inspection (see Section 8.11).

9 Conclusion

In this contribution we reviewed the digital tachograph system and addressed some conceptual vulnerabilities. Further directions for development are suggested.

References

1. Commission Regulation (EC) No 1360/2002 of 13 June 2002 adapting for the seventh time to technical progress. Council Regulation (EEC) No 3821/85 on recording equipment in road transport, Annex 1 B, Requirements for Construction, Testing, Installation and Inspection
2. Joint Interpretation Library (JIL): Security Evaluation and Certification of Digital Tachographs, JIL Interpretation of the Security Certification according to Commission Regulation (EC) 1360/2002, Annex 1B, Version 1.12, June 2003
3. ISO 16844-3, Road Vehicles – Tachograph Systems – Part 3: Motion Sensor Interface, 2004-11-01
4. Information Technology Security Evaluation Criteria (ITSEC), June 1991
5. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Part 2: Security functional requirements, Part 3: Security assurance requirements, January 2004, Version 2.2
6. BSI-PP-0002: Smartcard IC Platform Protection Profile, 1.0, available at `www.bsi.bund.de/cc/pplist/ssvgpp01.pdf`
7. PP/9806: Smartcard Integrated Circuit Protection Profile v2.0, available at `www.ssi.gouv.fr/site_documents/PP/PP9806.pdf`

8. PP9911: Common Criteria for IT Security Evaluation, Protection Profile: Smart card integrated circuit with embedded software, Version 2.0, EUROSMART, June 1999 – Registered by the French Certification Body under the reference PP/9911
9. Digital Tachograph System, European Root Policy, Version 2.0, Adminstrative Agreement 17398-00-12 (DG-TREN), European Commission
10. Ross J. Anderson, *On the Security of Digital Tachographs.* In: Lecture Notes in Computer Science, Vol. 1485, pp. 111
11. Ross J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, Inc., 2001
12. Ross J. Anderson, Invited Talk at ESCAR 2003

Secure In-Vehicle Communication

Marko Wolf[1], André Weimerskirch[2], and Christof Paar[1,2]

[1] Horst Görtz Institute (HGI) for IT Security,
Ruhr University of Bochum, Germany
{mwolf,cpaar}@crypto.rub.de

[2] escrypt GmbH, Bochum, Germany
{aweimerskirch,cpaar}@escrypt.com

Summary. This work presents a study of state of the art bus systems with respect to their security against various malicious attacks. After a brief description of the most well-known and established vehicular communication systems, we present feasible attacks and potential exposures for these automotive networks. We also provide an approach for secured automotive communication based on modern cryptographic mechanisms that provide secrecy, manipulation prevention and authentication to solve most of the vehicular bus security issues.

Keywords: automotive communication security, vehicular bus systems, LIN, CAN, FlexRay, MOST, Bluetooth

1 Introduction

Progress in automotive electronics proceeds unabated (Table 1). Today modern cars contain a multiplicity of controllers that are increasingly networked together by various bus communication systems with very different properties. Automotive communication networks have access to several crucial components of the vehicle, like breaks, airbags, and engine control. Moreover, cars that are equipped with driving aid systems like ESC (electronic stability control) or ACC (adaptive cruise control) allow deep interventions in the driving behavior of the vehicle. Further electronic drive-by-wire vehicle control systems will fully depend on the underlying automotive data networks. Although car communication networks assure safety against several technical interferences, they are mostly unprotected against malicious attacks. The increasing coupling of unsecured automotive control networks with new car multimedia networks like MOST (Media Oriented System Transport) or GigaStar as well as the integration of wireless interfaces such as GSM (Global System for Mobile Communications) or Bluetooth causes various additional security risks [32].

Table 1. Development of automotive electronics based on [30]

1970s	1980s	1990s	2000s
Electronic fuel injection	Electronic gearbox	Airbag	Drive-by-wire
Electronic control panel	Anti-lock brakes	Electronic navigation	Internet
Centralized door locking	Climate control	Electronic driving assistants	Telematics
Cruise control	Automatic mirror	Electronic traffic guidance	Ad-hoc networks
	Car phone	Voice control	Personalization

We begin in Section 2 by introducing respectively one well-known representative for each particular group of vehicular communication systems. We briefly describe technical properties of every representative (Section 2.3) and introduce two methods for vehicular bus interconnections (Section 2.4). Section 3 presents various exposures to automotive bus systems. We indicate possible attackers and present feasible attacks for each representative bus system. In the Section 4, we offer elementary approaches to improve automotive bus communication security along with a practical example implementation.

2 Automotive Bus Systems

2.1 Bus Communication

Unlike a point-to-point connection a bus is a communication system that can logically connect several peripherals, i.e. bus controllers over the same set of wires. The consequential potential savings of cost and weight encourage the increasing application of bus systems as communication systems within the automotive area. Moreover busses are easy to implement and to extend, and the failure of one node should not affect others. However, since in a bus system all nodes share the same communication line, they need schemes for collision handling or collision avoidance, or require a bus master which controls access to the shared bus resource. Furthermore, bus systems have a limited cable length and a limited number of nodes. The performance of a bus communication degrades the more nodes are connected, whereas a cable break can disable the entire vehicular bus network.

2.2 Vehicle Communication Systems

Today, a wide variety of vehicle communication systems are used in the automotive area. Possible applications range from electronic engine control, several driving assistants and safety mechanisms to the broad variety of infotainment applications. As shown in Table 2, we distinguish the following five different vehicle communication groups according to their essential technical properties and application areas. Local sub networks such as LIN (Local Interconnect

Table 2. Grouping of selected automotive bus systems

Subbus	Event-triggered	Time-triggered	Multimedia	Wireless
LIN	CAN	FlexRay	MOST	Bluetooth
K-Line	VAN	TTP	D2B	GSM
I^2C	PLC	TTCAN	GigaStar	Wi-Fi

Network) control small autonomous networks used for automatic door locking mechanisms, power-windows and mirrors as well as for communication with miscellaneous smart sensors to detect, for instance, rain or darkness. Event-triggered bus systems like CAN (Controller Area Network) are used for soft real-time in-car communication between controllers, networking for example the antilock breaking system (ABS) or the engine management system. Time-triggered hard real-time capable bus systems such as FlexRay, TTCAN (Time-Triggered CAN) or TTP (Time-Triggered Protocol) guarantee determined transmission times for controller communication and therefore can be applied in highly safety-relevant areas such as in most drive-by-wire systems. The group of multimedia bus systems like MOST, D2B (Domestic Digital Bus) and GigaStar arise from the new automotive demands for in-car entertainment that needs high-performance, wide-band communication channels to transmit high-quality audio, voice and video data streams within the vehicle. The wireless communication group contains modern wireless data transmission technologies that are increasingly expanding into the automotive area. They enable the internal vehicle network to communicate with other cars nearby, external base stations as well as the utilization of various location-based services. Figure 1 completes the overview with a short comparison of

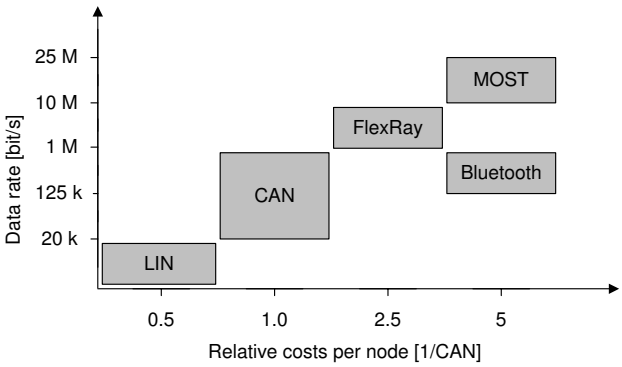

Fig. 1. Data rates and relative costs of automotive bus systems

typical data rate and relative cost per node for each vehicular communication group mentioned.

2.3 Bus Representatives

In the following, we give a short technical description of one appropriate representative from each identified vehicular communication network group (see Section 2). Further information can be found in [6, 10, 12, 22, 26].

LIN: The UART (Universal Asynchronous Receiver Transmitter) based LIN (Local Interconnect Network) is a single-wire sub network for low-cost, serial communication between smart sensors and actuators with typical data rates up to 20 kbit/s. It is intended to be used from the year 2001 everywhere in a car where the bandwidth and versatility of a CAN network are not required. A single master controls the collision-free communication with up to 16 slaves, optionally including time synchronization for nodes without a stabilized time base. LIN (similarly to CAN) is a receiver-selective bus system. Incorrectly transferred LIN messages are detected and discarded by the means of parity bits and a checksum. Besides the normal operation mode, LIN nodes also provide a sleep mode with lower power consumption, controlled by special sleep (or wake-up) message.

CAN: The all-round Controller Area Network, developed in the early 1980s, is an event-triggered controller network for serial communication with data rates up to one Mbit/s. Its multi-master architecture allows redundant networks, which are able to operate even if some of their nodes are defective. CAN messages do not have a recipient address, but are classified over their respective identifier. Therefore, CAN controllers broadcast their messages to all connected nodes and all receiving nodes and decide independently if they process the message. CAN uses the decentralized, reliable, priority-driven CSMA/CD (Carrier Sense Multiple Access/Collision Detection) access control method to guarantee the transmission of the top-priority message first. In order to employ CAN in the environment of strong electromagnetic fields, CAN offers an error mechanism that detects transfer errors, interrupts and indicates the erroneous transmissions with an error flag and initiates the retransmission of the affected message. Furthermore, it contains mechanisms for automatic fault localization including disconnection of the faulty controller.

FlexRay: FlexRay is a deterministic and error-tolerant high-speed bus, which meets the demands for future safety-relevant high-speed automotive networks. With its data rate of up to 10 Mbit/s (redundant single channel mode) FlexRay is targeting applications such as drive-by-wire and Powertrain. The flexible, expandable FlexRay network consists of up to 64 nodes connected point-to-point or over a classical bus structure. For physical transmission medium both optical fibers and copper lines are suitable. FlexRay is (similarly to CAN) a receiver-selective bus system and uses the cyclic TDMA (Time Division Multiple Access) method for data transmission control. Therefore, it

uses synchronous transmission for time-critical data and priority-driven asynchronous transmission for non-time-critical data via freely configurable, static and dynamic time segments. Its error tolerance is achieved by channel redundancy, a protocol checksum and an independent instance (bus guardian) that detects and handles logical errors.

MOST: The ISO/OSI standardized MOST (Media Oriented System Transport) serial high-speed bus became the basis for present and future automotive multimedia networks for transmitting audio, video, voice, and control data via fiber optic cables. The peer-to-peer network connects via plug-and-play up to 64 nodes in ring, star or bus topology. MOST offers, similarly to FlexRay, two freely configurable, static and dynamic time segments for the synchronous (up to 24 Mbit/s) and asynchronous (up to 14 Mbit/s) data transmission, as well as a small control channel. The control channel allows MOST devices to request and release one of the configurable 60 data channels. Unlike most automotive bus systems, MOST messages always include a clear sender and receiver address. Access control during synchronous and asynchronous transmission is realized via TDM (Time Division Multiplex) respectively CSMA/CA. The error management is handled by an internal MOST system service, which detects errors over parity bits, status flags and checksums and disconnects erroneous nodes if necessary.

Bluetooth: Originally developed to unify different technologies like computers and mobile phones, Bluetooth is a wireless radio data transmission standard in the license-free industrial, scientific, and medical (ISM) band at 2.45 GHz. It enables wireless ad-hoc networking of various devices like personal digital assistants (PDAs), mobile phones, laptops, PCs, printers, and digital cameras for transmitting voice and data over short distances up to 100 meters. Primarily designed as a low-cost transceiver microchip with low power consumption, it reaches data rates of up to 0.7 Mbit/s. Within the limited multi-master capable architecture, so-called Piconets, single Bluetooth devices can maintain up to seven point-to-point or point-to-multipoint connections. Bluetooth includes optional security mechanisms for authentication and confidentiality of messages at the link layer. Table 3 gives an overview of the characteristics of the five representative automotive bus systems.

2.4 Bus Interconnections

For network spanning communication, automotive bus systems require appropriate bridges or gateways to transfer messages among each other despite their different physical and logical operating properties. Gateways read and write all the different physical interfaces and manage the protocol conversion, error protection and message verification. Depending on their application area, gateways include sending, receiving and/or translation capabilities as well as some appropriate filter mechanisms. While so-called super gateways centrally interconnect all existing bus systems, local gateways link only two different

Table 3. Properties of selected automotive bus systems [14, 24, 5, 9, 17]

Bus	**LIN**	**CAN**	**FlexRay**
Adapted for	Low-level subnets	Soft real-time	Hard real-time
Target	Door locking	Antilock break system	Break-by-wire
application	Climate regulation	Driving assistants	Steer-by-wire
examples	Power windows	Engine control	Shift-by-wire
	Light, rain sensor	Electronic gear box	Emergency systems
Architecture	Single-master	Multi-master	Multi-master
Access	Polling	CSMA/CA	TDMA
control			FTDMA
Transfer	Synchronous	Asynchronous	Synchronous
mode			Asynchronous
Data rate	20 kbit/s	1 Mbit/s	10 Mbit/s
Redundancy	None	None	2 Channels
Error	Checksum	CRC	CRC
protection	Parity bits	Parity bits	Bus Guardian
Physical layer	Single-wire	Dual-wire	Dual-wire, Optical fiber
Security	None	None	None
	MOST	**Bluetooth**	
Adapted for	Multimedia	External communication	
Target	Entertainment	Telematics	
application	Navigation	Electronic toll	
examples	Information services	Internet	
	Mobile Office	Telediagnosis	
Architecture	Multi-master	Multi-master	
Access	TDM	TDMA	
control	CSMA/CA	TDD	
Transfer	Synchronous	Synchronous	
mode	Asynchronous	Asynchronous	
Data rate	24 Mbit/s	720 kbit/s	
Redundancy	None	79 Frequencies	
Error	CRC	CRC	
protection	System Service	FEC	
Physical layer	Optical fiber	Air	
Security	None	WEP	

bus systems together. Therefore, super gateways require some kind of sophisticated software and plenty of computing power in order to accomplish all necessary protocol conversions, whereas local gateways realize only the hard- and software conversion between two different bus backbones.

3 Exposures of Automotive Bus Systems

Ever since electronic devices were installed into cars, they have been a feasible target for malicious attacks or manipulations. Mileage counter manipulation [15, 16], unauthorized chip tuning or tachometer spoofing [1] are already common. Further possible electronic automotive applications like digital tachograph, electronic toll and electronic license plate or paid content and information services such as *Digital Rights Management* (DRM) or *Location Based Services* (LBS) increase the incentive for manipulating automobile electronics. Above all, unauthorized vehicle modifications can compromise particularly the driving safety of the respective car and of all surrounding road users. Besides

the most obvious attacker, the car owner, also garage employees (mostly on behalf of the car owner) and third parties such as competing manufacturers or other unauthorized persons and institutions may have incentives for attacks. Moreover, in contrast to most common computer networks, the car owner and the garage personnel have full physical access to all transmission media and affected devices of the automotive network. As the car owner normally has only low theoretical and technical capabilities, garage personnel and some external third parties may have both adequate background knowledge and the appropriate technical equipment, for feasible intrusions. This allows deep and above all permanent manipulation of the automobile electronics. Possible motivations of third parties for breaking into automotive networks may be attacks on the passenger's privacy (phone tapping, data theft) or well-directed attacks on particular vehicle components in the case of a theft or even a potential assault. Table 4 briefly represents the three groups of potential attackers and their respective capabilities. Apparently, technically sophisticated garage employees, acting on the owners instructions, are the most dangerous attacker group. Many analyses [2, 20, 21, 7] can verify the safety and reliability of ve-

Table 4. Attackers in the automotive area based on [18]

Attacker	**Capabilities**	**Physical access**
Car owner	Varied (generally low)	Full
Garage personnel	High	Full
Third party	Varied (may be high)	Feasible

hicle networks against random failures. Analyses that consider also intended malicious manipulations, i.e. discuss vehicular communication security, are still very rare [13, 23]. Thus, most existing automotive communication systems are virtually unsecured against malicious encroachments. Several factors make it difficult to implement security in the vehicular area. So far, safety has been the most crucial factor and therefore security has been only an afterthought. Automotive resource constraints, the multitude of involved parties and insufficient cryptographic knowledge cause additional difficulties when implementing appropriate precautions. Moreover, security may need additional hardware and infrastructures, may cause considerable processing delays and particularly generates extra costs, without apparent benefits. Nonetheless, vehicle electrification and in-car networking proceed unimpaired and the lack of security becomes an increasingly serious risk, so the emerging challenge in automotive communication is to provide security, safety and performance in a cost-effective manner.

Many typical characteristics of current automotive bus systems enable unauthorized access relatively easy. All communication between controllers is done completely unencrypted in plain text. Possible bus messages, their respective structures and communication procedures are specified in freely

available documents for most vehicle busses. Furthermore, controllers are not able to verify if an incoming message comes from an authorized sender at all. Nevertheless, the major hazard originates from the interconnection of all the car bus systems with each other. The net-spanning data exchange via various gateway devices, potentially allows access to any vehicular bus from every other existing bus system. In principle, each LIN, CAN or MOST controller is able to send messages to any other existing car controller. Hence, without particular preventive measures, a single comprised bus system endangers the whole vehicle communication network. In combination with the increasing integration of miscellaneous wireless interfaces, future attacks on automotive communication systems can be accomplished without contact, just by passing a car or via cellular phone from almost anywhere in the world. Breaking away the electronic mirror and connecting to the underlying LIN network with a mobile computer could already be a possible promising way to break into an expensive car today. In the next generation image-processing assistance for autonomous driving systems such as lane tracking or far field radar will access high safety-relevant vehicular driving systems based on information from external databases received via known, but quite insecure wireless links. Besides this, interconnections of multimedia busses like MOST and D2B, with the control network of the vehicle, enable software programs such as viruses or worms, received over inserted CD/DVDs, email messages or possibly attached computers, also to penetrate highly safety-relevant vehicular systems. Even if today modern gateways already include simple firewall mechanisms, most of them offer unprotected powerful diagnostic functions and interfaces that allow access to the whole car network without any restrictions. The consequences of successful attacks range from minor comfort problems to the the risk of an accident. Therefore, the probability of an attack and the level of security required in a given bus system depend on the potential consequences of loss or manipulation. As shown in Table 5, whereas attacks on LIN or multimedia networks may result in the failure of power windows or navigation software, successful attacks on CAN networks may result in malfunction of some important driving assistants that leads to serious impairments in driving safety. A successful systematic malfunction on real-time busses like FlexRay, which handle elementary driving commands like steering or breaking, can lead to acute hazards for the affected passengers and other surrounding road users. Nonetheless, also just a simple malicious car locking may have serious consequences for passengers [3]. In the following, we describe some feasible attacks on the protocol layer of the representative car bus systems described in Section 2. In doing so we assume we have either direct physical or logical access to the corresponding vehicle network. Physical access means a direct interconnection with the respective communication wires, whereas logical access means exploiting another (existing or deployed) controller or misusing the diagnosis or even a wireless interface [19].

Table 5. Endangerment of selected automotive bus systems

Group	*Subbus*	*Event-triggered*	*Time-triggered*	*Multimedia*	*Wireless*
Exemplar	LIN	CAN	FlexRay	MOST	Bluetooth
Exposure	Low	High	Acute	Low	Varied
Possible harms	Lessened functionality	Lessened driving safety	Risk of accident	Data theft, Lack of comfort	Unauthorized external access

LIN: Utilizing the dependency of the LIN slaves on their corresponding LIN master, attacking this single point of failure, will be a most promising approach. Introducing well-directed malicious sleep frames deactivates completely the corresponding subnet until a wake-up frame posted by the higher-level CAN bus restores the correct state again. The LIN synchronization mechanism can be another point of attack. Sending frames with bogus synchronization bytes within the SYNCH field makes the local LIN network inoperative or causes at least serious malfunctions. LIN is unprotected against forged messages.

CAN: The priority-driven CSMA/CD access control method of CAN network enables attacks that jam the communication channel. Constantly introduced topmost priority nonsense messages will always be forwarded first (even though they will be immediately discarded by the receiving controllers) and permanently prevent the transmission of all other CAN messages. Moreover, utilizing the CAN mechanisms for automatic fault localization, malicious CAN frames allow the disconnection of every single controller by posting several well-directed error flags. Furthermore, CAN is vulnerable to forged messages.

FlexRay: Similar to the CAN automatic fault localization, FlexRay's so-called bus guardian can be utilized for the well-directed deactivation of any controllers by appropriate faked error messages. Attacks on the common time base, which would make the FlexRay network completely inoperative, are also feasible, if within one static communication cycle more than f [1] malicious SYNC messages are posted into a FlexRay bus. Moreover, introducing well-directed bogus sleep frames deactivates corresponding power-saving capable FlexRay controllers. FlexRay is also vulnerable to forged messages.

MOST: Since in a MOST network one MOST device handles the role of the timing master, which continuously sends timing frames that allow the timing slaves to synchronize, malicious timing frames are suitable for disturbing or interrupting the MOST synchronization mechanism. Moreover, continuous bogus channel requests, which reduce the remaining bandwidth to a minimum, are a feasible jamming attack on MOST busses. Manipulated false bandwidth statements for the synchronous and asynchronous area within the boundary

[1] $f \geq n/3$, where n is the number of existing FlexRay nodes. Further reading in [31]

descriptor of a MOST frame can also make the network completely inoperative. Due to the utilized CSMA/CD access control method used within the asynchronous and the control channel, both are vulnerable to jamming attacks similar to CAN. MOST is also vulnerable to forged messages.

Bluetooth: Wireless interconnections imply a distinct security disadvantage over wired communications in that all information is broadcast over an open, easily tapping-capable air link. Although Bluetooth transmissions can be configured to be encrypted, there exist various feasible attacks [27, 4, 11]. Actually, even first worms and viruses begin infecting Bluetooth devices wirelessly [8, 29].

4 Approaches to Security

Many future vehicular applications will require high end-to-end communication security as enabling environment. It is then important that all transferred information can be seen and received in clear only by the desired parties, that potential modifications are impossible to conceal and that unauthorized parties are not able to participate in vehicular communication. Modern communication security mechanisms provide confidentiality, integrity and authentication based on cryptographic algorithms and protocols, to solve most of the car security problems. The uncontrolled interference of the vehicle communication networks can be prevented by a bundle of measures. In the following, we show three elementary practices to achieve vehicular bus communication security.

4.1 Controller Authentication

Authentication of all senders is needed to ensure that only valid controllers are able to communicate within automotive bus systems. All unauthorized messages may then be processed separately or are just immediately discarded. Therefore, every controller needs a certificate to authenticate itself against the gateway as a valid sender. A certificate consists of the controller identifier ID, the public key PK and the authorizations $Auth$ of the respective controller. The gateway in turn securely holds a list of public keys PK_{OEM} of all accredited OEMs (Original Equipment Manufacturers) of the respective vehicle. Each controller certificate is digitally signed by the OEM with its respective secret key SK_{OEM}. As shown in Table 6, the gateway again uses the corresponding public key of the OEM to verify the validity of the controller certificate. If the authentication process succeeds, the respective controller is added to the gateway's list of valid controllers.

4.2 Encrypted Communication

A fundamental step to improve the security of automotive bus communication is the encryption of all vehicular data transmission. Due to the particular

Table 6. Controller authentication

Authentication	
1.$Verify(Sig, PK_{OEM})$	Verify Sig with corresponding OEM public key PK_{OEM}
2.$ID, Auth$	Save controller properties, if verification succeeds
2.$C = E_{PK}(K_i)$	Send corresponding symmetric bus group key K_i

constraints of automotive bus communication systems (computing power, capacity, timing, ...), a combination of symmetric and asymmetric encryption meets the requirements on adequate security and high performance. Whereas fast and efficient symmetric encryption secures the bus-internal broadcast communication, asymmetric encryption is used to handle the necessary secure key distribution. In that case, all controllers of a local bus system share the same, periodically updated, symmetric key to encrypt their bus-internal communication. Asymmetric encryption provides the acquisition of the symmetric key for newly added authorized controllers and carries out the periodic symmetric key update, as well as the required authentication process. In our

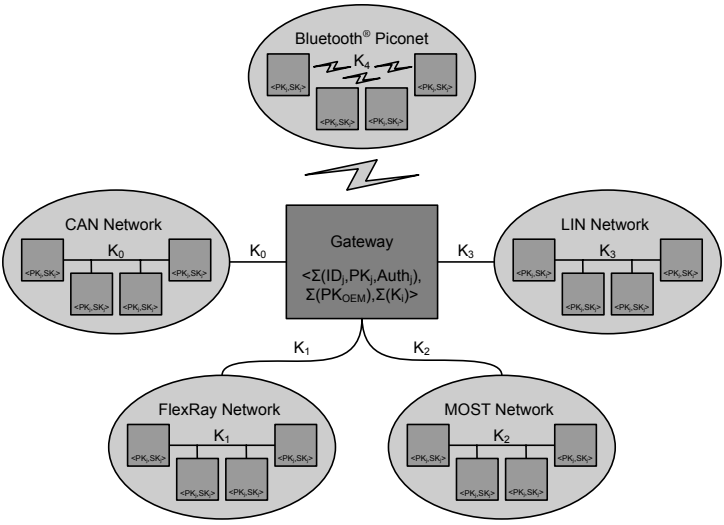

Fig. 2. Secure vehicular communication

example implementation shown in Fig. 2, a centralized super gateway processor connects all existing bus systems with each other. Therefore, all inter-bus communication is done exclusively only over the gateway processor. Moreover, the gateway has a protected memory area to securely store (tamper-resistant) the secret keys and the list of valid controllers together with their respective

authorizations $Auth$. The application of so-called trusted computing modules [28] can provide such particular secured memory portions. In our example, every successful verified bus controller holds the symmetric bus group key K_i as well as its own public and secret key pair PK_j, SK_j and the public key of the gateway PK_G. The gateway itself stores the certificates of each valid controller node as well as each bus-internal group key K_i for fast inter-bus communication. As all internal bus data is encrypted by K_i, only controllers that possess a valid K_i are able to decrypt and read all local broadcast bus messages. Since the centralized gateway holds the symmetric keys of every connected bus system, fast and secure inter-bus communication between valid controller nodes is provided. As shown in Table 7, every controller may optionally also receive a symmetric authentication key K_j from the gateway, to provide message integrity and sender authentication. If so, each controller could append a message authentication code (MAC) to every message, i.e. the respective hash value $H(M)$ encrypted with its personal authentication key K_j. Even though an asymmetric digital signature scheme could accomplish this task as well without additional authentication keys, it would probably exceed the timing requirements and the computing power of most automotive controllers. Table 8 shows the receipt of encrypted message C by a controller or

Table 7. Secured message sending with MAC authentication

Sending	
1. $C_1 = Enc(M, K_i)$	Encrypt message M with group key K_i
2. $MAC = Enc(H(M), K_j)$	Encrypt hash value of M with authentication key K_j
3. $C = C_1 \| \| MAC$	Send C composed of C_1 and MAC

the gateway processor. Whereas network internal controllers decrypt only the symmetric part C_1 of C, gateways have to verify also the optionally enclosed message authentication code MAC. Only if the sender verification succeeds and the sending controller has appropriate authorization does the gateway re-encrypt and forward the message into the targeted subnet. To enhance the

Table 8. Secured message receiving with MAC authentication

Receiving	
1. $M = Dec(C_1, K_i)$	Decrypt C_1 to message M with group key K_i
2. $H(M) \stackrel{?}{=} Dec(MAC, K_j)$	Verify integrity and sender of M with MAC (gateway only)
3. $Target \in Auth_j$	Forward M into target subnet if $Auth_j$ allow (gateway only)

security additionally, the gateway may initiate periodic bus group key updates. This prevents installing unauthorized controllers using a compromised

K_i or SK_j. To inform all controllers of a bus system, the gateway broadcasts for each controller on its current list of valid controllers a message encrypted with the respective public key PK_j of each controller. When every controller has decrypted its key update message with its secret private key SK_j, a final broadcast of the gateway may activate the new symmetric bus group keys.

4.3 Gateway Firewalls

For completing vehicular bus communication security, gateways should implement capable firewalls. If the vehicular controllers are capable of implementing MACs or digital signatures, the rules of the firewall are based on the authorizations given in the certificates of every controller. Therefore, only authorized controllers are able to send valid messages into (high safety-relevant) car bus systems. If the vehicular controllers do not have the abilities to use MACs or digital signatures, the rules of the firewall can be established only on the authorizations of each subnet. However, controllers of lower restricted networks such as LIN or MOST should generally be prevented from sending messages into high safety-relevant bus systems as CAN or FlexRay. Moreover, diagnostic functions and messages as well as all diagnostic interfaces, normally used only for analyses in garages or during manufacturing, should be disabled completely by authorized garage personnel to be inaccessible during normal driving operation.

5 Summary and Outlook

In this work, we have briefly presented current and future vehicular communication systems and pointed out several bus communication security problems. We presented an approach that uses modern communication security mechanisms to solve most of the local vehicular communication security problems. We expect that multimedia busses and wireless communication interfaces will soon be available in most modern automobiles. As already occurs now on the Internet, malicious attackers should not be underestimated and are most definitely a real existing threat. Even if a single successful attack causes only minor hazards for passengers it may seriously jeopardize public confidence in a brand [25]. Since future automotive systems and business models particularly depend on comprehensive and efficient measures that provide vehicular communication security, adequate technical, organizational and financial expenditures have to be arranged today.

References

1. Ross J. Anderson. On the security of digital tachographs. In *ESORICS '98: Proceedings of the 5th European Symposium on Research in Computer Security*, pages 111–125, London, UK, 1998. Springer-Verlag.
2. M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, M. Peri, and S. Pezzini. Fault-tolerant platforms for automotive safety-critical applications, 2003.
3. Bangkok Post. Computer traps thailand's finance minister Suchart. *Bangkok Post*, May 19 2003.
4. Bundesamt für Sicherheit in der Informationstechnik. Bluetooth – Gefährdungen und Sicherheitsmaßnahmen, 2003.
5. CAN in Automation. Website. `www.can-cia.org`, 2005.
6. T. Dohmke. Bussysteme im Automobil CAN, FlexRay und MOST. Master's thesis, TU Berlin, March 2002.
7. Richard Evans and Jonathan D. Moffett. Derivation of safety targets for the random failure of programmable vehicle based systems. In *SAFECOMP*, pages 240–249, 2000.
8. F-Secure. Bluetooth worm Cabir. `www.f-secure.com/v-descs/cabir.shtml`, 2004.
9. FlexRay Group. FlexRay main specifications. `www.flexray.com`, 2005.
10. H. Heinecke, A. Schedl, J. Berwanger, M. Peller, V. Nieten, R. Belschner, B. Hedenetz, P. Lohrmann, and C. Bracklo. FlexRay – ein Kommunikationssystem für das Automobil der Zukunft. *Elektronik Automotive*, September 2002.
11. Markus Jakobsson and Susanne Wetzel. Security weaknesses in bluetooth. In *CT-RSA 2001: Proceedings of the 2001 Conference on Topics in Cryptology*, pages 176–191, London, UK, 2001. Springer-Verlag.
12. R. Kraus. Ein Bus für alle Fälle. *Elektronik Automotive*, January 2002.
13. Andreas Lang and Jana Dittmann. Steigende Informationstechnologie: Sicherheitsrisiko im Fahrzeugbau. In Erhard Plödereder, Hubert Keller, Hans von Sommerfeld, Peter Dencker, Michael Tonndorf, and Francesca Saglietti, editors, *Automotive – Safety & Security 2004 -Ũ Sicherheit und Zuverlässigkeit für automobile Informationstechnik*, LNI, pages 21–33, Aachen, September 2004. GI, Shaker Verlag.
14. LIN Consortium. LIN main specifications. `www.lin-subbus.de`, 2005.
15. Maximilian Maurer. Tachomanipulation – Jungbrunnen aus dem Laptop. *ADAC Presseservice*, March 2005.
16. Mosen Automobilelektronik. Company website. `www.tachoteam.de`, 2005.
17. MOST Cooperation. MOST main specifications. `www.mostnet.org`, 2005.
18. Christof Paar. Eingebettete Sicherheit im Automobil. In *Embedded Security in Cars Conference*, Cologne, Germany, November 2003. GITS AG.
19. Jan Pelzl and Thomas Wollinger. Security Aspects of Mobile Communication Systems. In *this book*.
20. M. Plankensteiner. Sicherheit beim Bremsen und Lenken. *Elektronik Automotive*, September 2002.
21. S. Poledna, G. Stöger, R. Schlatterbeck, and M. Niedersüß. Sicherheit auf vier Rädern. *Elektronik Automotive*, October 2001.
22. M. Randt. Bussysteme im Automobil. ECT Workshop Augsburg, 2002.

23. Maxim Raya and Jean-Pierre Hubaux. The security of vehicular networks. Technical report, Laboratory for Computer Communications and Applications (LCA), School of Computer and Communication Sciences, EPFL, Switzerland, March 2005.
24. Robert Bosch GmbH. CAN main specifications. `www.can.bosch.com`, 2005.
25. A. Rother. *Krisenkommunikation in der Automobilindustrie - Eine inhaltsanalytische Studie am Beispiel der Mercedes-Benz A-Klasse*. PhD thesis, Neuphilologische Fakultät, Universität Tübingen, November 2003.
26. B. Rucha and G. Teepe. LIN - local interconnect network. *Elektronik Automotive*, January 2003.
27. Yaniv Shaked and Avishai Wool. Cracking the bluetooth PIN. `www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/index.html`, 2005.
28. Trusted Computing Group. TPM main specifications. `www.trustedcomputinggroup.org`, 2005.
29. Unknown. Handyviren - Der Ernstfall wird wahrscheinlicher. *Spiegel Online*, 2004.
30. U. Weinmann. Anforderungen und Chancen automobilgerechter Softwareentwicklung. In *3. EUROFORUM-Fachkonferenz*, Stuttgart, Germany, July 2002.
31. J.L. Welch and N. Lynch. A new fault-tolerant algorithm for clock synchronization. In *Information and Computation*, volume 77 of *Information and Computation*, pages 1–36, April 1988.
32. T. Zeller and N. Meyersohn. Can a virus hitch a ride in your car? *New York Times*, March 13 2005.

A Survey of Research in Inter-Vehicle Communications

Jun Luo and Jean-Pierre Hubaux

School of Computer and Communication Sciences
EPFL, CH-1015 Lausanne, Switzerland
{jun.luo, jean-pierre.hubaux}@epfl.ch

Summary. As a component of the *intelligent transportation system* (ITS) and one of the concrete applications of mobile ad hoc networks, *inter-vehicle communication* (IVC) has attracted research attention from both academia and industry of, notably, US, EU, and Japan. The most important feature of IVC is its ability to extend the horizon of drivers and on-board devices (e.g., radar or sensors) and, thus, to improve road traffic safety and efficiency. This chapter surveys IVC with respect to key enabling technologies ranging from physical radio frequency to group communication primitives and security issues. The mobility models used to evaluate the feasibility of these technologies are also briefly described. We focus on the discussion of various MAC protocols that seem to be indispensable components in the network protocol stack of IVC. By analyzing the application requirements and the protocols built upon the MAC layer to meet these requirements, we also advocate our perspective that ad hoc routing protocols and group communication primitives migrated from wired networks might not be an efficient way to support the envisioned applications, and that new coordination algorithms directly based on MAC should be designed for this purpose.

1 Introduction

Inter-vehicle communication (IVC), on one hand, is an important component of the *intelligent transportation system* (ITS) architecture. It enables a driver (or its vehicle) to communicate with other drivers (or their vehicles) that locate out of the range of *line of sight* (LOS) (or even out of radio range if a multihop network is built among several vehicles). As a result, information gathered through IVC can help improve road traffic safety and efficiency. On the other hand, moving vehicles equipped with communication devices form an instance of long-envisioned mobile ad hoc networks [25]. Benefiting from the large capacities (in terms of both space and power) of vehicles, the nodes of these networks can have long transmission ranges and virtually unlimited lifetimes. Also, many existing protocols designed for ad hoc networks and experiences learned from the related research can be applied. One of the

earliest studies on IVC was started by JSK (Association of Electronic Technology for Automobile Traffic and Driving) of Japan in the early 1980s. Later, well-known research results on platooning[1] have been demonstrated by the California-based PATH project [13] and the Chauffeur EU project [12]. The cooperative driving systems of Japan in the late 1990s and 2000 (e.g., DEMO 2000 [34]) exhibit another set of important applications of IVC. A related topic is *adaptive cruise control* (ACC). Traditional solutions to this issue involve mainly automatic control systems for individual vehicles [35], but IVC can help to make the coordination more efficient. Recently, the transmission of information about incidents, emergencies, or congestion from (a) preceding vehicle(s) to vehicles following behind also became an important application of IVC (e.g., [24]). The newly initiated European Project CarTALK 2000 [27] tries to cover problems related to safe and comfortable driving based on IVC. It focuses on the design, test and evaluation of co-operative driver assistance systems by taking into account both IVC and *road-to-vehicle communication* (RVC), where RVC is used to provide vehicles with access to fixed networks [23]. CarTALK 2000 also co-operates with other projects such as the German FleetNet [9] and NOW (Network on Wheels: `www.network-on-wheels.de`) for the development of IVC.

The main applications of IVC, as summarized by [27], can be roughly categorized into three classes:

- **Information and warning functions**: Dissemination of road information (including incidents, congestion, surface condition, etc.) to vehicles distant from the sites of interest.
- **Communication-based longitudinal control**: Exploiting the "look-through" capability of IVC to help avoiding accidents and to arrange platooning.
- **Co-operative assistance systems**: Coordinating vehicles at critical points such as blind crossings (a crossing without light control) and highway entries.

There are also "added value" applications, such as location-based services and multiplayer games. Considering the tight coupling between a specific application and its supporting mechanisms, we will not devote a section to describe applications, but rather mention applications when their enabling mechanisms are discussed. The remainder of this chapter is structured as follows. Section 2 discusses the radio bands used in IVC physical layer. Section 3 details various proposals for IVC MAC. Section 4 presents several routing protocols dedicated to IVC. Section 5 describes application of group communication in IVC. Section 6 discusses security issues. Section 7 briefly describes different mobility models used in IVC simulations. Finally, Section 8 makes conclusions.

[1] Platooning is by definition the technique of coupling two or more vehicles together electronically to form a train. This means that the total headway for vehicles going in the same direction could be reduced, and the capacity of the road would consequently be increased.

2 Radio Frequency Spectrum

In this section, we discuss the frequency spectra used by different IVC systems; we do not address technical issues such as the antenna and modulation in the physical layer. As the media for the IVC, both infrared and radio waves have been studied and employed for experimental systems. The radio waves include VHF, micro, and millimeter waves. The communication with infrared and millimeter waves are within the range of LOS and usually directional, whereas those with VHF and microwaves are of broadcast type. Although VHF waves such as the 220 MHz band have been used because of their long communication distance, the mainstream nowadays is microwaves. The *dedicated short-range communication* (DSRC) in the USA, allocated by FCC, spans over 75 MHz of the spectrum in the 5.9 GHz band. In Japan, 5.8 GHz DSRC was used by DEMO 2000 and 60 GHz millimeter wave has been tested to evaluate its performance under the hidden terminal situation. In Europe, Chauffeur chose 2.4 GHz at the beginning; it also changed to 5.8 GHz later. CarTALK/FleetNet chose UTRA TDD because of the availability of an unlicensed frequency band at 2010–2020 MHz in Europe. It is worth noting that infrared, in spite of its various drawbacks, has been adopted by most projects including JSK, PATH, and CarTALK, typically for co-operative driving.

3 MAC/PHY Layer: (W)LAN vs. 3G

Currently, there are two main approaches in developing wireless MAC for IVC. They differ in the adopted radio interface. One approach is based on existing wireless LAN physical layers, such as that of IEEE 802.11 or Bluetooth. An alternative approach is to extend 3G cellular technology, i.e., CDMA, for decentralized access. The advantage of the first approach is its inherent support for distributed coordination in ad hoc mode, but the flexibility of radio resource assignment and of transmission rate control is low. On the contrary, 3G extensions have the potential of high granularity for data transmission and flexible assignment of radio resources due to the CDMA component, but suffer from the complexity of designing coordination function in ad hoc mode. We now discuss these two approaches separately.

3.1 WLAN Extension

Although it is possible to use WLAN standards directly for RVC [23], the outcome might not be satisfactory for IVC since, for example, these mechanisms are designed without having mobility in mind. Migrating a WLAN technology for vehicular applications requires development in the following areas:

a. Resistance to potentially more severe multipath effects
b. Time synchronization between nodes susceptible to move rapidly

c. Distributed resource allocation in a network of highly dynamic topology.

While (a) depends much on the development of hardware and proper physical layer, there are proposals that tried to solve (b) and (c) solely within the MAC layer. We hereafter discuss several proposals that inherit certain parts of the existing standards but try to solve some aforementioned aspect(s) by adding new features. Lee et al. [16] from PATH suggest the use of a token ring protocol similar to IEEE 802.4 to solve the contention of radio resources. The protocol includes the mechanism to construct, recover, join, and leave a ring, as well as the token circulation and multiple token resolution in the ring. Although this protocol is claimed to be adaptive to dynamic topology and rely only on the physical layer of IEEE 802.11, the performance evaluations did not take mobility into account and the protocol evaluated is implemented on top of IEEE 801.11 DCF. Therefore, a convincing proof is still necessary to show that this protocol is suitable for IVC. Katragadda et al. [15] propose a Location-based Channel Access (LCA) protocol. Assuming the availability of location-aware devices with each node, the LCA protocol divides a geographical area into cellular structure with each cell having a unique channel associated with it. Within a given cell, any multiple access schemes, including CSMA, CDMA, and TDMA, can be used. In this sense, LCA is not simply an extension of WLAN. Considering the similarity between LCA and the spatial division multiple access (SDMA) in traditional cellular networks, a doubt may be raised about the protocol's adaptability to high mobility scenarios like in IVC. There are other proposals based on some traditional LAN technologies such as the non- or p-persistent CSMA used by DOLPHIN [33]. The contribution of this work is to show that the non-persistent CSMA outperforms the p-persistent one regarding packet loss in those cases usually involved in IVC. As a result, the non-persistent CSMA is adopted as the IVC protocol of the DEMO 2000 co-operative driving [34].

Numerous proposals are concerned with modifying IEEE 802.11 for some specific case(s). We do not discuss them here due to their minor significance to IVC.

3.2 3G Extension

It is impossible to directly apply 3G technologies, because they are designed for cellular networks, which are inherently centralized. The following problems have to be addressed in order to extend 3G technologies for IVC:

a. Distributed radio resource management
b. Power control algorithms
c. Time synchronization.

All these problems are due to the absence of centralized infrastructure. Therefore, the solution should rely on distributed media access control. Many proposals suggest using Reservation ALOHA (R-ALOHA) for distributed channel

assignment. R-ALOHA has higher throughput than slotted-ALOHA, since a node that catches a slot can use it in subsequent frames as long as it has packets to send. However, there are two problems to be solved in order to make traditional R-ALOHA work for IVC. On one hand, R-ALOHA has a potential risk of instability in the case of many participating nodes and frequent reservation attempts due to short packet trains. Lott et al. [17] solve this problem by letting every node reserve a small part of transmit capacity permanently even if it has no packets to send. This results in a circuit-switched broadcast connection primarily used for signaling purposes. The time synchronization is built upon the information from GPS and additional synchronization sequence in parallel to data transmission. Further system evaluation under high node mobility can be found in [28]. On the other hand, traditional R-ALOHA needs a broadcast environment for all nodes to receive all the transmitted signals and, most important, to get the status information of slots. Since IVC suffers from the hidden terminal problem, destructive interference with already established channels can occur and accessing nodes have no idea about the outcome of their transmission. To overcome these problems, Borgonovo et al. [4] have studied a new protocol, named Reliable R-ALOHA (or RR-ALOHA). This protocol transmits additional information to let all nodes be aware of the status of each slot, thus safely allowing the same reservation procedure of R-ALOHA to occur in IVC. The two-hop relaying that propagates the status information is very similar to what is used in ad hoc routing to let a node know the neighbor information of its neighbors. However, since this work is very recent and is still under study, no field test or simulation results are reported, leading to the question about its performance under high mobility networks. Both protocols are based on UTRA TDD, which is chosen by CarTALK/FleetNet as the target system. Several MAC protocols for ad hoc networks combine CDMA with random channel access (e.g., [30]). These protocols usually start their transmission immediately, irrespectively of the state of the channel. Under appropriate code assignment and spreading-code schemes, *primary collisions* (i.e., two nodes with the same code try to access the channel together) can be avoided. However, Muqattash and Krunz [21] pointed out that RA-CDMA (*random access* CDMA) suffers from *multi-access interference* (MAI), resulting in *secondary collisions* (also known as *near–far problem* in the literature) at a receiver. As a consequence, CA-CDMA [21] uses a modified RTS/CTS reservation mechanism. The channel is split into control and data channels. RTS/CTS is transferred over control channels to let all potentially interfering nodes be aware of the channel status. In contrast to IEEE 802.11, interfering nodes may be allowed to transmit concurrently, depending on some criteria. The protocol also exploits knowledge of the power levels of the overheard RTS/CTS to perform power control that intends to alleviate the near–far problem. According to the simulation results (especially the comparison between CA-CDMA and IEEE 802.11), this protocol is a quite promising MAC for ad hoc networks, but simulations (or even field tests) that take mobility into account are necessary to justify its deployment in IVC.

Summary

Although a number of MAC protocols have been proposed, more efforts are needed to put them into practice. Currently, IEEE 802.11b is still the one used for demonstration [10], and IEEE 802.11a is chosen by ASTM (American Society for Testing and Materials) to be the basis for its standard of DSRC [1]. However, the MAC protocol based on UTRA TDD, promoted by CarTALK, could be another promising solution for IVC (at least in the EU).

4 Network Layer: The Role of Location Awareness

Almost all unicast routing protocols proposed for IVC are position-based. Basically, any existing position-based routing protocol for ad hoc networks [31] can be applied to IVC, but the protocols can be optimized by taking into account the special features of vehicles. For example, GPS, Geographic Information System (GIS), and digital maps can help a node to be aware of its location and the surroundings, such as the road topology. Since the road topology has a strong influence on the network topology in IVC, this knowledge does help to make the routing protocol more efficient [32, 7]. Furthermore, one of the most recent results on position-based routing [11] proposes a forwarding scheme avoiding the need of beacons for improved efficiency. One of the real implementations, demonstrated by FleetNet [10] (see also [20]), has not exploited these special features of vehicles yet. Their protocol behaves like a reactive routing protocol by requesting the location of a destination when sending a packet. Then greedy geographical forwarding is used to forward packets. We also notice that most people try to solve the problem of unicast routing just because "it is challenging in ad hoc networks". Actually, considering the applications mentioned in Section 1 (which involve more group-oriented rather than pairwise communications), we are really wondering if unicast routing still has the same significance as in "general" ad hoc networks. The application of broadcasting is usually to disseminate traffic information. Most solutions suggest scoped-flooding for broadcasting. Thanks to the peculiarity of this application, certain optimizations can be applied. For example, Wischhof et al. [36] adaptively change the inter-transmission interval according to the significance of the event conveyed by the message in transmission, while Briesemeister et al. [6] use a randomized interval. If the locations of vehicles are again taken into consideration, a multiresolution data structure can be used to express information in the message [19]. The intuition here is that the further a vehicle is from the event, the less detail it needs.

Summary

Considering the application requirements for IVC, broadcast/geocast routing that disseminates information to a set of nodes that are located in the neighborhood seems to be a necessary mechanism; it could be optimized according

to the requirements of an application. On the contrary, unicast routing might be superfluous in most cases.

5 Group Communication: Promising but Unattended Research Area

Although two of the main applications of IVC, namely platooning and co-operative driving, imply the need for group communication, researchers seldom pay attention to this area. While broadcast protocols mentioned in the previous section perform group-oriented information dissemination, group communication primitives would still be welcome for IVC, because reliability could be important in certain critical situations. We hereby discuss a few related works and try to envision some potential research aspects. Briesemeister [5] suggests reducing the group membership service to the local environment of a node, because of the impossibility result of primary-component group membership in asynchronous systems with crash failures (which is the situation with IVC). The *localized group membership service* (LGMS) only tracks the membership of neighbors and installs a local view at each node. Obviously, the views of different nodes differ from each other. Although LGMS provides an interesting solution to the problem that the author aims at, i.e., congestion area detection, its weak properties (e.g., no agreement on the membership) make it hard to apply to a broad context. Actually, this service does not support any functions with a reliability requirement due to the lack of global view of the group. Gorman [22] raises a very interesting problem about coordinating vehicles at a blind crossing, which he terms 4-way stop (4WS) problem, and tries to apply group communication to perform coordination functions. While the problem itself is intriguing since it is an important aspect of co-operative driving, the proposed solution needs further improvement. It is not yet clear whether all the properties mentioned in the thesis, which are direct migrations from traditional group communication, could really work in IVC environment. Some researchers from the theoretical area of distributed computing also noticed the importance of applying group communication in IVC. Meier and Cahill [18] proposed an event-based middleware to support group-oriented applications. They focus on small groups that are apparently abstracted from scenarios in IVC.[2] However, the underlying membership service that attempts to locate all nodes in a given geographical area is a bit costly (in terms of communication consumption), and it is not clear if applications really need this kind of membership service. Baehni et al. [2] consider the problem of sharing certain resources among a group of vehicles. They propose an algorithm that solves the problem in a synchronous model. Another important contribution is to prove the impossibility of achieving fairness and concurrency as well as the impossibility of solving the problem in an asynchronous model.

[2] Unfortunately, they implement their experiments only in an RVC scenario.

Summary

Group communication is definitely an important component of IVC, but it has seldom been addressed. Existing proposals show that potential design considerations could include: (i) building the system directly upon the MAC layer and (ii) tracking membership in a more lightweight way than a global tracking.

6 Security: An Emerging Research Topic

Security of IVC has been ignored so far by the research community. The only publication we could find is by El Zarki et al. [37]. The paper proposes a system called DAHNI (Driver Ad Hoc Networking Infrastructure), to be mounted (in the long run) on each vehicle. DAHNI includes both processing and wireless communication facilities, allowing each car to constitute a local communication area around itself. In this way, each car can exchange vital signs with the neighboring vehicles. The authors discuss the security implications of such a solution. One of their conclusions is a bit surprising: they mention that no confidentiality is needed, thereby neglecting the tremendous privacy concerns that such a solution is likely to raise. They mention that no key distribution is necessary, which is true for the scenarios they consider; but if vehicles need to securely estimate the distance between them, the establishment of symmetric keys is required. In [14], we have shown that the wireless identification of vehicles is likely to rely more and more on *electronic licence plates*. We have identified the attacks against such a scheme, including those against the privacy of vehicle drivers; we have sketched appropriate techniques to thwart them. We have shown that this principle enables fundamental mechanisms such as location verification; it also supports secure distance estimation. Finally, we have explained how these mechanisms can support cooperative driving. More recently, we have proposed a security architecture that is compliant with the constraints of privacy preservation [26].

7 Mobility Model: Basis of Protocol Simulation

The mobility pattern underlying an inter-vehicle network is quite different from the "random waypoint" model that is intensively used for ad hoc network simulations. Fortunately, researchers of applied mathematics have already proposed many tools for traffic modeling (e.g., [3] provides a survey of these approaches), which can be used to extend network simulators such as *ns*-2 and GloMoSim. Note that the simulations for MAC protocols of IVC must also take mobility into account [28], which is not necessarily the case for the traditional MAC protocol (even wireless MAC like IEEE 802.11). Usually, mathematical modeling for traffic can be classified into three categories [3],

according to the phenomenological observation of the system: (i) microscopic modeling, (ii) statistical description, and (iii) macroscopic description. We are not going to give details about each method, but rather provide examples where certain protocols are simulated. Microscopic modeling is suitable for simulating group communications, because the applications of these protocols are often concerned with local behaviors of vehicles. For example, Briesemeister applies a microscopic model in her thesis [5], which describes the velocity and position of each vehicle at a given time. Many other papers discussing routing protocols use macroscopic models where the mobility pattern is defined by four parameters: average vehicle speed v in m/s, traffic density ρ in vehicles/km, traffic flow q in vehicles/s, and net time gap τ in seconds. Usually, assumptions are made on two of them since the other two can be calculated subsequently. For example, Rudack et al. [29] assume a v of normal distribution and a τ of exponential distribution, while Briesemeister et al. [6] assign uniform distribution for both v and ρ. All the aforementioned models deal with one-dimensional cases, but the real mobility pattern of a vehicle is in a two (even three) dimensional space. To this purpose, the cellular automaton approach [8], combined with road patterns created based on certain maps, is adopted by FleetNet to simulate their Self-Organizing Traffic Information System (SOTIS) [36]. This approach is based on Markov chain theory to emulate the vehicles' behavior at a cross road.

Summary

The application context has to be taken into account when choosing a mobility model to evaluate certain protocols.

8 Conclusion

Various aspects of IVC are surveyed in this chapter. The chaper shows that the design of communication protocols in the framework of IVC is extremely challenging due to the variety of application requirements and the tight coupling between an application and its supporting protocols. Most existing proposals are concerned with MAC and routing protocols. While MAC is definitely an important component of the IVC protocol stack, we are not convinced that routing protocols are necessary in most cases, as they are supposed to be in general ad hoc networks. In many situations, especially those related to co-operative driving, local but distributed coordination functions sitting directly upon MAC would be more efficient solutions. In addition, since vehicles will become "smarter", partially due to the installation of IVC systems, security and privacy are becoming new concerns that both academia and industry should pay attention to. Finally, mathematical models for road traffic are important tools in developing IVC systems, because simulations are still necessary in testing large-scale communication systems.

References

1. ASTM E 22123-02. Telecommunications and information exchange between roadside and vehicle systems. In *ASTM International*, 2001. `www.astm.org`.
2. S. Baehni, R. Baldoni, B. Pochon, and R. Guerraoui. The driving philosophers. Technical Report IC/2004/15, EPFL, 2004.
3. N. Bellomo and M. Delitala. On the mathematical theory of vehicular traffic flow I: Fluid dynamic and kinetic modelling. *Mathematical Models and Methods in Applied Sciences*, 12(2):1801–1843, 2002.
4. F. Borgonovo, A. Capone, M. Cesana, and L. Fratta. ADHOC MAC: A new, flexible and reliable MAC architecture for ad-hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'03)*, 2003.
5. L. Briesemeister. *Group membership and communication in highly mobile ad hoc networks.* PhD thesis, School of Electrical Engineering and Computer Science, Technical University of Berlin, 2001.
6. L. Briesemeister, L. Schafers, and G. Hommel. Dissemination messages among highly mobile hosts based on inter-vehicle communication. In *Proc. of IEEE Intelligent Vehicle Symposium (IV'00)*, 2000.
7. A. Cheng and K. Rajan. A digital map/GPS based routing and addressing scheme for wireless ad hoc networks. In *Proc. of IEEE Intelligent Vehicle Symposium (IV'03)*, 2003.
8. B. Chopard, P.O. Luthi, and P.-A. Queloz. Cellular automata model of car traffic in a two-dimensional street network. *Journal of Physics A: Mathematical and General*, 29(10):2325–2336, 1996.
9. W. Franz, R. Eberhardt, and T. Luckenbach. Fleetnet – internet on the road. In *Proc. of the 8th World Congress on Intelligent Transportation Systems (ITS'01)*, 2001. `www.fleetnet.de`.
10. H. Fubler, H. Hartenstein, W. Franz, W. Enkelmann, M. Moske, and C. Wagner. The Fleetnet demonstrator. In *Demos of the 9th ACM/IEEE international conference on Mobile Computing and Networking (MobiCom'03)*, 2003.
11. H. Fubler, J. Widmer, M. Kasemann, M. Mauve, and H. Hartenstein. Contention-based forwarding for mobile ad-hoc networks. *Elsevier's Ad-Hoc Networks*, 1(4):351–369, 2003.
12. O. Gehring and H. Fritz. Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication. In *Proc. of the 1st IEEE Conference on Intelligent Transportation System (ITSC'97)*, pages 117–122, 1997.
13. J.K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6):21–32, 1994.
14. J.-P. Hubaux, S. Čapkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy Magazine*, 2(3), 2004.
15. S. Katragadda, G. Murthy, R. Rao, M. Kumar, and R. Sachin. A decentralized location-based channel access protocol for inter-vehicle communication. In *Proc. of the 57th IEEE Semiannual Vehicular Technology Conference (VTC'03 Spring)*, 2003.
16. D. Lee, R. Attias, A. Puri, R. Sengupta, S. Tripakis, and P. Varaiya. A wireless token ring protocol for intelligent transportation systems. In *Proc. of the IEEE Intelligent Transportation System Conference (ITSC'01)*, 2001.

17. M. Lott, R. Halfmann, E. Schulz, and M. Radimirsch. Medium access and radio resource management for ad hoc networks based on UTRA TDD. In *Proc. of the 2nd ACM/SIGMOBILE Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'01)*, 2001.
18. R. Meier and V. Cahill. Exploiting proximity in event-based middleware for collaborative mobile applications. In *Proc. of the 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'03), LNCS 2893*, 2003.
19. L.B. Michael. Adaptive layered data structure for inter-vehicle communciation in ad-hoc communcation networks. In *Proc. of the 8th World Congress on Intelligent Transportation Systems (ITS'01)*, 2001.
20. M. Moske. Real-world evaluation of a vehicular ad hoc network using position-based routing. Master's thesis, Department of Computer Science, University of Mannheim, 2003.
21. A. Muqattash and M. Krunz. CDMA-based MAC protocol for wireless ad hoc networks. In *Proc. of the 4nd ACM/SIGMOBILE Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'03)*, 2003.
22. Eoin O'Gorman. Using group communication to support inter-vehicle coordination. Master's thesis, Department of Computer Science, University of Dublin, 2002.
23. J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11 for "Automobile" users. In *Proc. of the 23rd IEEE INFOCOM*, 2004.
24. C. Passmann, C. Brenzel, and R. Meschenmoser. Wireless vehicle to vehicle warning system. In *SAE 2000 World Congress*, 2002.
25. C. Perkins, editor. *Ad hoc networking*. Addison-Wesley, 2001.
26. M. Raya and J.-P. Hubaux. Security Aspects of Inter-Vehicle Communications. In *Proc. of the Swiss Transport Research Conference (STRC)*, 2005.
27. D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz. CarTALK 2000 – safe and comfortable driving based upon inter-vehicle-communication. In *Proc. of the IEEE Intelligent Vehicle Symposium (IV'02)*, 2002. `www.cartalk2000.net`.
28. M. Rudack, M. Meincke, K. Jobmann, and M. Lott. On traffic dynamical aspects inter-vehicle communication (IVC). In *Proc. of the 57th IEEE Semi-annual Vehicular Technology Conference (VTC'03 Spring)*, 2003.
29. M. Rudack, M. Meincke, and M. Lott. On the dynamics of ad-hoc networks for inter-vehicle communications. In *Proc. of the International Conference on Wireless Networks (ICWN'02)*, 2002.
30. E. Sousa and J.A. Silvester. Spreading code protocols for distributed spread-spectrum packet radio networks. *IEEE Transactions on Communications*, 36(3):272–281, 1988.
31. I. Stojemenovic. Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):138–134, 2002.
32. J. Tian, L. Han, and K. Rothermel. Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks. In *Proc. of the IEEE Intelligent Transportation System Conference (ITSC'03)*, 2003.
33. K. Tokuda, M. Akiyama, and H. Fujii. DOLPHIN for inter-vehicle communications system. In *Proc. of IEEE Intelligent Vehicle Symposium (IV'00)*, 2000.

34. S. Tsugawa, K. Tokuda S. Kato, T. Matsui, and H. Fujii. An overview on DEMO 2000 cooperative driving. In *Proc. of the IEEE Intelligent Vehicle Symposium (IV'01)*, pages 327–332, 2001.
35. A. Vahidi and A. Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Trans. on Intelligent Transportation Systems*, 4(3), 2003.
36. L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann. Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication. In *Proc. of IEEE Intelligent Vehicle Symposium (IV'03)*, 2003.
37. M. El Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian. Security issues in a future vehicular network. *European Wireless*, 2002.

Part III

Embedded Security Technologies

Fundamentals of Symmetric Cryptography

Sandeep Kumar and Thomas Wollinger

Horst Görtz Institute (HGI) for Security in Information Technology,
Ruhr University of Bochum, Germany
{kumar,wollinger}@crypto.rub.de

Summary. It is widely recognized that data security will play a central role *not only* in the design of future IT systems, but also in all kind of systems in which electronic data are exchanged. Cryptology is the main tool to realize data security. Cryptographic primitives will not only secure the data communication, but will provide safety and reliability of the given system. The latter is sometimes far more important for certain applications which involve automated control based on the data communication between different devices. Cryptology provides two different kinds of algorithms, namely symmetric and asymmetric (public-key) algorithms.

This chapter gives an introduction to symmetric key cryptography and its subgroups – block ciphers and stream ciphers. We also provide short descriptions of the most commonly used algorithms in industry: DES and AES. We will focus on their special properties from an implementation point of view. Major concentration will be on software and hardware implementations of DES, 3-DES, AES and different modes of operations of block ciphers so that they can be used also as stream ciphers.

1 Introduction

It is widely recognized that data security will play a central role *not only* in the design of future IT systems, but also in all kinds of systems in which electronic data are exchanged. Until a few years ago, only computers and data transferred through the Internet had been protected against someone who wants to harm the system. Nowadays, there has been a shift towards security critical applications realized on embedded systems and we find that the *bad* third party is also interested in manipulating or preventing the functionality of these systems. Many of those applications rely heavily on security mechanisms, including security for wireless phones, faxes, wireless computing, pay-TV, and copy protections schemes for audio/video consumer products and digital cinema. However, we will find in the near future more and more protection mechanism in daily applications, like smart windows, refrigerators, traffic signs and cars. Note that security mechanisms will not only secure the data communication, but will provide safety and reliability of the given system.

The latter applies to many applications, e.g. the traffic signs and cars, even more important than the secrecy of data, because it is a protection against failure by chance.

It is also important to note that a large share of those embedded applications will be wireless, which makes the communication channel especially vulnerable and the need for security even more obvious.

The merging of the transmission of data and computation functionality for encryption requires data processing in real time. This will be one of the challenges for future applications because they are mostly realized with embedded processors. Hence, for development there are only limited resources available. On the other side, however, we need to perform extensive computations to be able to provide some of the security mechanisms. Another crucial issue is the performance of algorithms. One needs encryption algorithms to run at the transmission rates of the communication links. Slow–running cryptographic algorithms can translate into malfunction of systems.

The explosive growth of digital data that is transmitted also brings additional security challenges. Millions of bits are sent over wire or wireless each day. In the future, megabytes of sensitive data (e.g. secret documents but also control information) will be transferred and moved over communication channels around the world. Thus, it is imperative that all these transactions be realized in a secure manner. Specifically, unauthorized access to information must be prevented, privacy must be protected, and authenticity must be established. Cryptography, or the art and science of keeping messages secure [19], allows us to solve these problems.

The engineer who designs a system has also to consider physical security using different platforms. An encryption algorithm running on a general-purpose computer has only limited physical security, as the secure storage of keys in memory is difficult on most operating systems. On the other hand, hardware encryption devices can be securely encapsulated to prevent attackers from tampering with the system. Thus, custom hardware is the platform of choice for security protocol designers. Hardware solutions, however, come with the well-known drawback of reduced flexibility and potentially high costs. These drawbacks are especially prominent in security applications in which one needs to change the cryptographic primitives.

Many of the new security protocols decouple the choice of cryptographic algorithm from the design of the protocol. Users of the protocol negotiate on the choice of algorithm to use for a particular secure session. Hence, these applications must support many cryptographic algorithms and protocols. In addition, they need to be "algorithm agile," that is, able to select from a variety of algorithms (e.g. IPSec – the security standard for the Internet). However, it is obvious that one can integrate these kinds of protocols for all applications because of limited resources. Hence, the security engineer will always have to decide how much security is necessary and what resources are available.

In this chapter we will first give an overview (Section 2), differentiating different security services and how symmetric key cryptography can solve some

of them. Section 3 explains symmetric key algorithms and its subgroups: block ciphers and stream ciphers. The following two sections, Sections 4 and 5, concentrate on two specific symmetric key algorithms, DES and AES, describing their structure and different implementation possibilities. Sections 6 and 7 describe the different modes of operation and other cryptographic primitives that can be built from block ciphers respectively. We end this contribution with some conclusions.

2 Overview

Cryptography involves the study of mathematical techniques that provide the following security services:

- *Confidentiality* is a service used to keep the content of information accessible to only those authorized to have it. This service includes both protection of all user data transmitted between two points over a period of time as well as protection of traffic flow from analysis.
- *Integrity* is a service that requires that computer system assets and transmitted information be capable of modification only by authorized users. Modification includes writing, changing, changing the status, deleting, creating, and the delaying or replaying of transmitted messages. It is important to point out that integrity relates to active attacks and therefore, it is concerned with detection rather than prevention. Moreover, integrity can be provided with or without recovery.
- *Authentication* is a service that is concerned with assuring that the origin of a message is correctly identified. That is, information delivered over a channel should be authenticated as to the origin, date of origin, data content, time sent, etc. For these reasons this service is subdivided into two major classes: entity authentication and data origin authentication. Notice that the second class of authentication implicitly provides data integrity.
- *Non-repudiation* is a service which prevents both the sender and the receiver of a transmission from denying previous commitments or actions.

Symmetric cryptography provides the ability to securely and confidentially exchange messages between two parties. This is especially important if the data should not be revealed to any third party. Integrity can be guaranteed by using the proper *mode of operation* with symmetric cipher. Authentication without non-repudiation can also be achieved if the secret key is known only to the two parties. Asymmetric algorithms have much more fascinating properties which will be discussed in the next chapter [24].

3 Symmetric Key Cryptography

Symmetric key cryptographic algorithms are the basic building blocks of any secure systems which require confidentiality. They are used normally to encrypt messages in bulk that are transmitted between two systems. In these kinds of cryptographic algorithms, the keys used for encryption and decryption are the same for both the communicating entities and hence called a *symmetric cipher*. It can be regarded as a safe-box inside which messages can be put and then locked and sent to the other party. If the other party has the key to the lock, then the party can open and read the messages in the safe-box. The security of the symmetric cipher depends only on the key which is known only to these two parties (the algorithm is assumed to be public), and so these ciphers are also sometimes referred to as *private-key algorithms*. The exchange of these keys between the parties may have to be done using a different secure channel or by using public–key crypto–systems, which will be discussed in the next chapter.

Since symmetric key algorithms are used to encrypt the bulk of the data, they have to run at high speeds or at least at the bandwidth of the channel so as not to cause a bottleneck. There has been a lot of study to make symmetric key cryptography as efficient as possible without compromising the security.

Symmetric key algorithms are mainly divided into two categories: *block ciphers* and *stream ciphers*.

3.1 Block Ciphers

Block ciphers encrypt the messages in data blocks of fixed length, mostly 64 bits or 128 bits. The most well known block ciphers are the Data Encryption Standard (DES) [16], and the Advanced Encryption Standard (AES) [21].

DES was the first commercially standardized block cipher with 64-bit data block size and 56-bit key size. The algorithm has been widely used in different industries ranging from the banking sector to Internet security. Its widespread use has been largely due to the fact that it was the only standardized and openly available algorithm outside the domain of military and secret agencies that had been extensively studied by the cryptanalytic community. The design criteria for the DES algorithm and its security analysis were not released in the public domain which caused many to distrust the design process. There have been no major weaknesses found in the algorithm to date to practically break it other than the relatively small size of the key. This allows a brute force attack running through all the keys becoming easily attractive with decreasing cost for the computational power [6]. DES finally expired as a US standard in 1999, out living its recommended usage life, and the National Institute of Standards (NIST) selected the Rijndael algorithm as the AES in October 2000. In the transition phase Triple-DES was approved as an FIPS standard [15].

The AES selection process was remarkably different from that of its predecessor by making it as an open challenge to the cryptographic research community. The algorithm Rijndael [4] developed by Daemen and Rijmen was finally selected from a large set of algorithms submitted to the AES challenge running over three years. AES [21] supports block size of 128 bits and variable key sizes of 128 bits, 192 bits, and 256 bits to give a choice of different security levels based on its application. It is important to point out that the trend in modern block cipher design has been to optimize the algorithms for efficient software implementations, in contrast to DES, which was designed for hardware.

There have been other block ciphers like IDEA which have been popular to a certain extent because it was used as the default algorithm for the Pretty Good Privacy (PGP) email encryption package. Most of these algorithms were developed because of the security concerns of the small key size which was finally fixed with the AES. So most applications which demand security have switched to AES (including PGP and SSL).

For practical reasons, however, DES continues to be used widely in legacy systems and versions like Triple-DES (especially in the banking community) are still standard. These versions counter the disadvantage of the smaller key length by encrypting the message multiple times with DES with different keys.

It should be noted that block ciphers are only the building blocks for confidentiality. They can be used in different ways to achieve this purpose, called *modes of operation*. Some of the well–known modes are Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter mode [1, 14, 10, 11]. These will be discussed in Section 6 later in this chapter.

3.2 Stream Ciphers

The main idea behind stream ciphers is the *one-time pad (OTP)* [22] encryption (also called Vernam cipher), which is the only known cipher to have a provably unconditional security [20]. The OTP works by bitwise XOR of the plain-text with a one-time key which is of the same length. The problem of having a secret key of the same length as the message to be transmitted makes OTP encryption inconvenient in practice. This shortcoming is overcome by using a pseudo-random generator (but unconditional security holds no more). Today's stream ciphers operate on a single bit of plain text (or a few bytes of data) being XORed to a pseudo-random key stream generated based on a master key and an initialization vector. This is also a secret key encryption, i.e. both parties need to know the secret master key.

Stream ciphers are especially useful in situations where transmission errors are highly probable because they do not have error propagation. In addition, they can be used when the data must be processed one symbol at a time because of lack of device memory or limited buffering. Mobile telephony is one such situation where stream ciphers are used because they are inherently

much simpler in design and are very suitable for mobile devices which are constrained by the chip area and power.

A5 cipher is the most well–known stream cipher as it is used in "Global System for Mobile communication (GSM)" telephony (discussed in "Security Aspects of Mobile Communication Systems" in this book). It was the first time that cryptographic algorithms were widely used in mass market equipment. There are different versions of the A5: European version A5/1 and a weakened version A5/2 for export. The algorithm is built into mobile phones and was kept a secret; however, there have been open implementations by reverse engineering. A5/1 uses a 64–bit cipher key but most implementations were found to be using an artificially weakened 54-bit key by zeroing the top 10 bits. There have been different attacks published which makes A5 completely weak for present security needs. In September 2003, a new algorithm A5/3, also called Kasumi (a block cipher), has been standardized by the "3rd Generation Partnership Project (3GPP)" for the new third generation (UMTS) mobile networks (discussed in "Security Aspects of Mobile Communication Systems" in this book). Kasumi is a modified version of Mitsubishi's MISTY [12] algorithm, which has been widely studied by the open cryptographic community. The A5/3 algorithm uses a 128-bit key but the effective key length is only 64 bits due to the way the key is derived.

The other most widespread stream cipher is the RC4 cipher developed by Ron Rivest of RSA security in 1987. RC4 is also not openly available and is kept as a trade secret of RSA security. There has been an alleged copy of RC4 source code called ARC4 which was published anonymously in 1994 and numerous cryptanalyses based on it. Though RC4 by itself can have an arbitrary key size, because of export restrictions a 40-bit key was normally used but now 128 bits is common. It is used in different applications like SSL, Adobe Acrobat and Microsoft Office.

Another stream cipher that is used widely is E0 in Bluetooth (discussed in "Security Aspects of Mobile Communication Systems" in this book). It has a maximum key length of 128 bits but the actual key size used is negotiated between the communicating devices.

It is worth mentioning that the recent European project "New European Schemes for Signatures, Integrity, and Encryption (NESSIE)", whose main purpose was to recommend cryptographic schemes to provide different security services and be included in standards and products, has decided to recommend none of the submitted stream ciphers because none of the algorithms met the rather stringent security requirements put forward by the project [17, 18].

In April 2005, a new call for stream cipher primitives was launched by "European Network of Excellence for Cryptology (ECYRPT)" which is expected to find a suitable candidate for widespread adoption.

It should be noted that stream ciphers can also be built from block ciphers by using the different chaining modes (discussed in Section 6), though it might be inefficient in terms of area but adds no extra cost if a block cipher is

already implemented on the system. Hence block ciphers like AES are being used increasingly to implement the key stream generator for stream ciphers.

4 Data Encryption Standard

Data Encryption Standard has been the first and most well–known standard for secure data storage and mail systems, electronic fund transfers and electronic business data interchange. It has been in use since 1976 when it was developed by IBM for securing documents and standardized [16] by the National Bureau of Standards.

DES is a block cipher which operates on 64-bit blocks of plain text data and uses a 56-bit key as a secret. Essentially the same algorithm works both for encryption and decryption. Like every block cipher, the DES algorithm also consists of a repeating process called a *round*. The round is executed sixteen times (see Fig. 1) to generate the cipher text. The key is also expanded using a *key expansion* routine (Fig. 2) to allow a different sub-key to be used for each round. A detailed description of the implementation can be found in [19, 14].

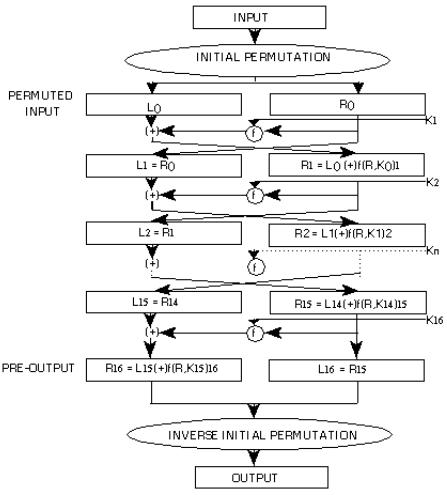

Fig. 1. Transformation order for DES

DES is broadly based on the two important properties suggested for block ciphers by Shannon in his ground breaking paper [20]: *diffusion* and *confusion*. Diffusion is the property of spreading out the influence of a single plain text digit over many cipher text digits so as to hide the statistical structure of the plain text. Confusion means using the enciphering transformations that complicates the determination of how the statistics of the cipher text depends

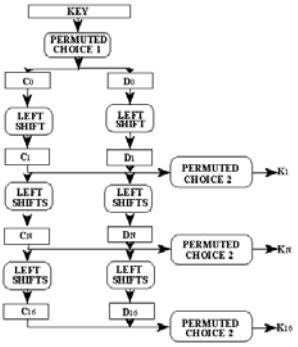

Fig. 2. DES Key expansion

on the statistics of the plain-text. Diffusion and confusion are realized in each of the rounds by using permutations and substitutions of the plain text as shown in the Fig. 3. The sixteen rounds help to propagate the confusion and diffusion characteristics.

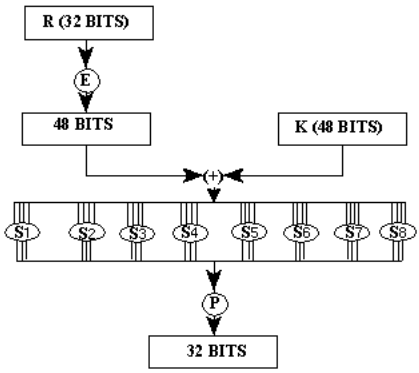

Fig. 3. DES Round function

DES when developed was designed to be implemented efficiently in hardware. This allows very compact and fast implementations of DES possible in hardware (see Table 1). But software implementations have normally been hard because specific bits within a byte are addressed, which is hard to realize in software efficiently.

Table 1. Hardware Performance of DES

DES Encryption and Decryption				
	Virtex-II FPGA		ASIC TSMC 180nm	
	Size (Slices)	Data Rate (Mbps)	Size (gates)	Data Rate (Mbps)
Compact	527	128	7.9K	266
Standard	803	240	11.8K	533
Fast	1,367	430	21.8K	1,067
Ultra Fast	4,305	1,941	56.7K	4,267

4.1 Bit Slice implementation of DES

Eli Biham suggested the bit slice implementation [3] of DES to overcome the problems with selecting single bits within the block. The data blocks to be encrypted are arranged along their bits and the first bit of each data item is selected for encrypting the data in registers as shown in the Fig. 4. The CPU can be viewed as 8, 16 or 32-bit parallel processors based on the word size of the processors. The processor then acts on the data as an SIMD instruction. This also allows encrypting multiple blocks of DES in parallel.

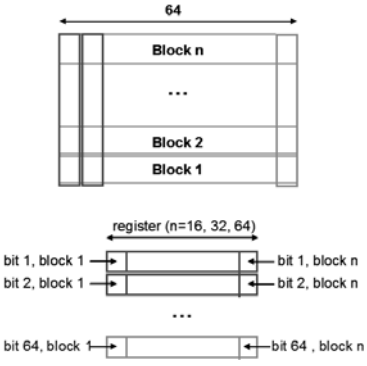

Fig. 4. Bit slice implementation

Since DES operates bit-wise, all of the operations match into a normal implementation of DES and do not affect the final computation bits of each stage. The only difficulty can be the non-linear S-boxes because the S-box inputs are 48 bits of data which have to be converted to a different set of S-box which gives the same output. Biham shows that such an S-box can be easily implemented as algebraic functions which are implemented as gates, and then converted to basic instructions on the processor. Though it might take much longer for running the substitution stage, the overall speed–up in

the DES implementation overcomes the extra overhead, giving a more efficient implementation.

4.2 Triple DES

Multiple encryption on data using different sets of secret keys provides some extra security but has to be done more thoughtfully to avoid a man-in-the-middle attack [14]. FIPS 46-3 [15] defines a standardized way to implement Triple-DES (also referred as Triple Data Encryption Algorithm TDEA) based on ANSI X9.52 [2]. Triple-DES encryption is defined as a compound operation of DES encryption (E) and decryption (D) operations with three different keys as $C = E_{K3}(D_{K2}(E_{K1}(P)))$ where P is the plain text and C is the cipher text. Triple-DES decryption is similarly defined as $P = D_{K1}(E_{K2}(D_{K3}(C)))$. The standard also specifies the following keying options for the key bundle (K1,K2,K3):

- Keying Option 1: K1, K2 and K3 are independent keys.
- Keying Option 2: K1 and K2 are independent and K3=K1.
- Keying Option 3: K1=K2=K3.

5 Advanced Encryption Standard

The new Advanced Encryption Standard was selected by NIST after more than three years open competition by the cryptographic community to replace the DES standard. Finally a set of five finalist AES candidates were chosen from the ones submitted by the research community and the industry. These were vigorously tested by a committee not only to confirm its practical security but also its ease in implementation both in embedded devices and also on fast servers. NIST and leading cryptographers from around the world found that all five finalist algorithms had a very high degree of security. But Rijndael, developed by Belgian cryptographers Joan Daemen and Vincent Rijmen, was selected as the AES [21] because it had the best combination of security, performance, efficiency, implementability and flexibility.

The AES is a 128-bit block cipher with three possible key lengths: 128, 192 and 256 bits. The number of rounds also depends on the key length. A detailed description about the inherent security and design principles can be found in the book [5] from Daemen and Rijmen.

AES consists of four invertible transformations which provide the confusion and diffusion required by Shannon's principle:

- Byte Substitution
- Shift Rows
- Mix Columns
- Add Round Key

We give here only a short description of the different transformations for AES-128. The interested reader can find more detailed information in [5]. Initially the input data is divided into 128-bit data block, which are arranged as bytes in a 4 by 4 matrix called a **State**. The keys are similarly arranged in a matrix format of 4 by keylength/32. Then the various transformations are performed in the order as shown in the Fig. 5.

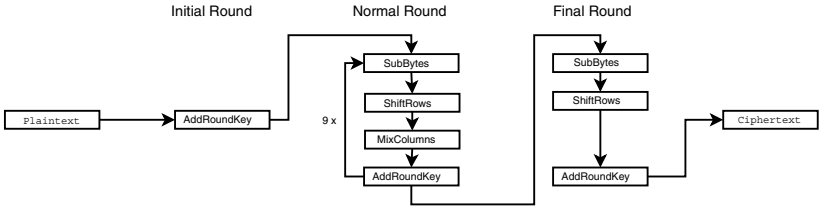

Fig. 5. Transformation order for AES-128 encryption

Byte Substitution

This is a nonlinear, invertible byte substitution using the so called S-Box (see Fig. 6). Two transformations are performed on each of the bytes independently:

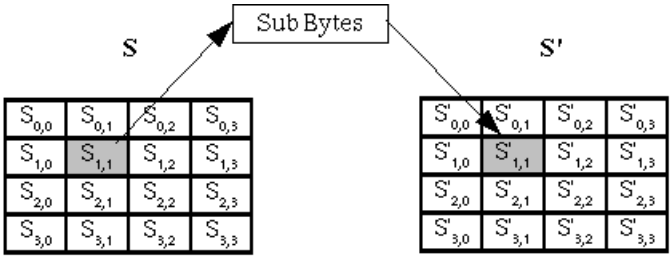

Fig. 6. Byte Substitution

- First each byte is substituted by its multiplicative inverse in $GF(2^8)$, and for the special case the element {00} is mapped to itself.
- Then the following affine transformation over $GF(2)$ is applied:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Shift Rows

As Fig. 7 depicts, the *Shift Rows* operation cyclically shifts each row of the **State** by a certain offset. The first row is not shifted at all, the second row is shifted by one, the third row by two and the fourth row by three bytes to the left.

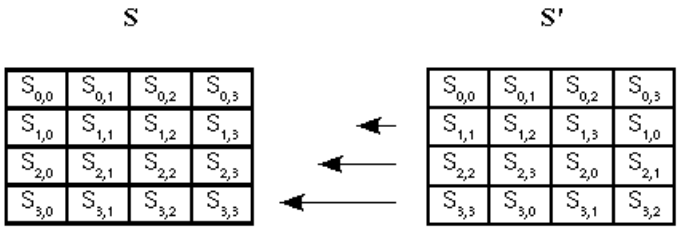

Fig. 7. Shift rows

Mix Columns

The columns of the **State** are processed one at a time during this operation. The bytes are interpreted as coefficients of a four-term polynomial over $GF(2^4)$. Each column is multiplied modulo $x^4 + 1$ with a fixed polynomial $\mathrm{a(x)} = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. This can be written as the following matrix multiplication, where $s'(x) = a(x) \otimes s(x)$:

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \text{for } 0 \leq c \leq 3.$$

As one can see in Fig. 8 the columns of the **State** are processed independently of one another.

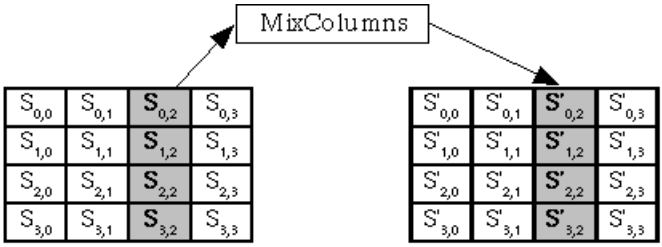

Fig. 8. Mix columns

Add Round Key

This operation adds the 128-bit *round key* generated from *Key Expansion* to the 128-bit **State**. It is a simple XOR-addition of the *roundkey* and the **State**.

Key Expansion

For a complete AES-128 encryption or decryption 10 *round keys* are needed. The *Key Expansion* derives them from the initial key iteratively as it is depicted in Fig. 9. The operation involves rotation and substitution on four byte words. A complete description can be found in [5]. It is important to note here that the key expansion requires only the previous four bytes of the round key to generate the next round key.

Decryption

In decryption mode, the inverse operations are used in the reverse order as shown in Fig. 10. Details of each transformation can be found in [5].

5.1 Implementation

We describe here efficient implementation of the AES for different platforms. This is an evolving area and the best results can be found only by following the latest research (for example [9, 23, 8]).

Software implementation of AES

AES was chosen such that it is easy to implement both on low–end 8-bit processors as well as high–end 32-bit processors. This is evident if one looks at the performance of the AES on different platforms (Table 2). The internal AES operations can be broken down into 8-bit operations, which is important because many cryptographic applications run on constrained devices such as

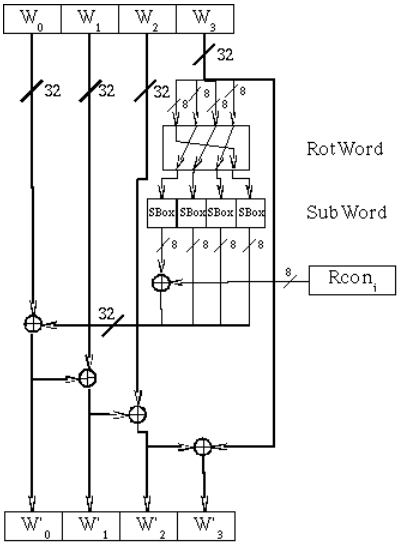

Fig. 9. Key Expansion

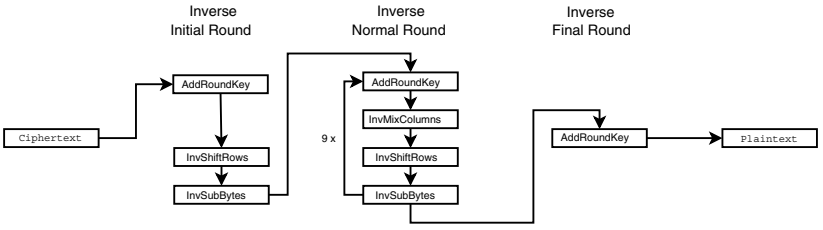

Fig. 10. Transformation order of the AES-128 decryption

8-bit microcontrollers and smart cards. Furthermore, one can combine certain steps to get a suitable performance in the case of 32-bit platforms.

Implementation on a low–end 8-bit processor is straightforward since AES is defined at the byte level. Therefore the different transformations can be implemented directly as described in the specification. The *Sub Bytes* is implemented using a 256-byte table-lookup. The *Shift Row* operation is a simple reordering of bytes. The *Mix Column* stage is effectively implemented on 8-bit processors by realizing `02` and `03` of the matrix in the polynomial basis as $02 = x$ (simple shift with possible reduction) and $03 = x + 1$ (simple shift as before with addition). More information on an 8-bit implementation can be found in [4].

Implementation for a 32-bit processor requires a reordering of the *Sub Byte* and *Shift Rows* of the round operations. This is possible due to the algebraic properties of the operations. This reordering allows us to come up

with a different set of substitution boxes called the *T-boxes* which are much bigger in size (four $256 * 4$-bytes) but are not too big for storage on high–end processors. This allows the complete round to be implemented as 16 table lookups and gives a very efficient implementation also for a 32-bit processor. A good description of the method can be found in [4].

Table 2. Software Performance for AES-128

Processor Type	Throughput (Mbit/s)	Clock (MHz)
8051 Chip card[13]	0.024	3.7
68HC705 Chip card	0.067	5
ARM	2.49	28.65
DSP TI TMS320C6201	112.3	200
Power PC	152.9	500
Athlon	535.7	1400
Alpha ACP21264	581.3	1000
Pentium III	718.4	1330

Hardware implementation of AES

AES is also amenable to very fast hardware implementations, since the operations that it performs on the data to be encrypted can be mapped very efficiently at the bit level. AES has been very efficiently implementable over a wide spectrum of hardware resources and speed constraints ranging from very fast processing to very small sizes. We give here some values both from research papers and commercially available products to show their applicability to hardware implementations.

Differences between AES and DES

The biggest difference between AES and DES is that AES is not a Feistel cipher [7] where in each round only half the data bits were operated upon. In AES, the whole data block is equally transformed to get a very high amount of diffusion and confusion. The second difference is the fact that the S-boxes used in AES are based on more open algebraic properties compared to the heuristical S-boxes used in DES, which led many to suggest possible back doors known only to NIST. The other big difference is with the key schedule operation in AES. Compared to DES where the key schedule involved just shifting of bits, in AES the S-boxes are used to derive the round keys, which gives it a more non-linear structure. On the implementation side, AES is more easier to implement in software compared to DES because it handles data byte-wise. DES, on the other hand, was inherently easier to be built in hardware because of its bit–wise operations which are not at word boundaries.

Table 3. Hardware performance of AES-128

AES Encryption				
	Virtex-II FPGA		ASIC TSMC 180nm	
	Size (Slices)	Data Rate (Mbps)	Size (gates)	Data Rate (Mbps)
Compact	403+4BRAM	350	14.8K	581
Standard	696+4BRAM	250-341	18.2K	426-581
Fast	573+10BRAM	1,323	27K	2,327
Ultra Fast	2181+100BRAM	10,880	203K	25,600
AES Decryption				
Compact	549+4BRAM	290	16.4K	581
Standard	746+4BRAM	290-426	19.2K	426-581
Fast	778+10BRAM	1,064	34K	2,327
Ultra Fast	3998+100BRAM	9,344	283K	25,600

6 Modes of Operation for Block Ciphers

Most of the time block ciphers are used in a chained way for better security and to prevent any replay attacks. FIPS 81 [1] specifies four different modes of operation for DES: Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB). ECB is a straightforward implementation without any chaining. CBC is the most commonly used chaining method used with DES and many other block ciphers.

ANSI X9.52 [2] specifies the seven different modes of operation to be used with Triple-DES: TDEA Electronic Codebook (TECB), TDEA Cipher Block Chaining (TCBC), TDEA Cipher Block Chaining – Interleaved (TCBC-I), TDEA Cipher Feedback (TCFB), TDEA Cipher Feedback – Pipelined (TCFB-P), TDEA Output Feedback (TOFB), and TDEA Output Feedback Mode of Operation – Interleaved (TOFB-I). The TECB, TCBC, TCFB and TOBF modes are based upon the ECB, CBC, CFB and OFB modes respectively, obtained by substituting the DES encryption/decryption operation with the Triple-DES encryption/decryption operation. Some of the modes are backward compatible with DES with third keying option [15] used in Triple-DES.

NIST also specifies five different modes of operation for AES at present and will propose new ones in future. The present specification [10] suggests ECB, CBC, CFB, OFB and the Counter(CTR) mode.

Considering the different modes of operation is important because depending on the mode different methodologies might be used to enhance the throughput of a block cipher. For example, pipelining might be used in hardware implementations (bit slicing method in software) of block cipher operating in ECB mode, whereas this technique cannot be used when using CBC, CFB, or OFB modes of operation.

7 Other Uses of Block Ciphers

Block ciphers can also be used to generate other cryptographic primitives without requiring any extra hardware or software costs. For example, some of the modes of operation can be used to create a stream cipher by reducing the number of data bits used in each block to a smaller value. This creates a big overhead in terms of processing but saves on extra hardware or software required to implement an extra cryptographic primitive and its associated weakness sometimes due to wrong implementations. AES was suggested to be used as a stream cipher in OFB mode or the Filter Counter Mode, in which case the key stream is generated by encrypting some counter using a secret key [4].

Another cryptographic primitive that can be implemented is the Message Authentication Codes (MAC). CBC-MAC has been the most common method of creating a MAC from a block cipher by running it in Chain Block Cipher mode. But there are newer versions of generating MACs which are more secure, like EMAC and RIPE-MAC [14].

Symmetric key algorithms can also be used for other functions like pseudo-random number generators [4] which are very often required in security devices to randomly choose certain parameters.

8 Conclusions

The US–based NIST, "New European Schemes for Signatures, Integrity and Encryption (NESSIE)" and the Japanese "Cryptographic Evaluation Project for the Electronic Government (CRYPTREC)" have conducted in the past few years open studies on different symmetric ciphers suggested both by the research community and industry. Most of these ciphers evaluated are good both in terms of security and implementation properties. This whole set can provide developers with a wide choice of algorithms with different key lengths on which security services can be built. The current minimum key length for high security is assessed to be 80 bits and hence single DES should definitively not be used anymore. It is, however, always recommended to use the most commonly used ciphers like AES because they are more widely tested for any open vulnerabilities. Sticking to a common standardized algorithm also allows easy compatibility and extension between different communicating devices in the future without any additional costs. AES is soon emerging as this de facto standard among different security protocols as the best choice for the block cipher at present.

References

1. DES Modes of Operation, FIPS, Federal Information Processing Standard, Pub No. 81. Available at `csrc.nist.gov/fips/change81.ps`, December 1980.
2. American National Standards Institute. *ANSI X9.52-1998, Triple Data Encryption Algorithm Modes of Operation*, 1998. Available at `webstore.ansi.org/ansidocstore/dept.asp?dept_id=80`.
3. E. Biham. A Fast New DES Implementation in Software. In *Fourth International Workshop on Fast Software Encryption*, LNCS 1267, pages 260–272, Berlin, Germany, 1997. Springer-Verlag.
4. J. Daemen and V. Rijmen. AES Proposal: Rijndael. In *First Advanced Encryption Standard (AES) Conference*, Ventura, California, USA, 1998.
5. Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES – the Advanced Encryption Standard*. Springer-Verlag, Berlin, Germany, 2002.
6. Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly & Associates, Inc., 103a Morris Street, Sebastopol, CA 95472, USA, Tel: +1 707 829 0515, and 90 Sherman Street, Cambridge, MA 02140, USA, Tel: +1 617 354 5800, July 1998.
7. H. Feistel. Cryptography and Computer Privacy. *Scientific American*, (228):15–23, 1973.
8. Marc Joye and Jean-Jacques Quisquater, editors. *Cryptographic Hardware and Embedded Systems – CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*. Springer, 2004.
9. Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.
10. M. Dworkin. *NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation – Methods and Techniques*. National Institute of Standards and Technology, US Department of Commerce, December 2001. Available at `csrc.nist.gov/encryption/tkmodes.html`.
11. M. Dworkin. *Draft NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation: The RMAC Authentication Mode – Methods and Techniques*. National Institute of Standards and Technology/U.S. Department of Commerce, November 4, 2002. Available at `csrc.nist.gov/encryption/tkmodes.html`.
12. Mitsuru Matsui. New block encryption algorithm MISTY. In Eli Biham, editor, *Fast Software Encryption: 4th International Workshop*, volume 1267 of *Lecture Notes in Computer Science*, pages 54–68, Berlin, 1997. Springer-Verlag.
13. Gael Hachëz, François Koeune, and Jean-Jacques Quisquater. cAESar results: Implementation of Four AES Candidates on Two Smart Cards. In *Proceedings: Second AES Candidate Conference (AES2)*, Rome, Italy, March 1999.
14. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, USA, 1997.
15. National Institute of Standards and Technology, US Department of Commerce. *Federal Information Processing Standards FIPS PUB 46-3, Data Encryption Standard (DES)*, October 25, 1999. Available at `csrc.nist.gov/CryptoToolkit/tkencryption.html`.

16. NIST FIPS PUB 46-3. *Data Encryption Standard*. Federal Information Processing Standards, National Bureau of Standards, US Department of Commerce, 1977.
17. B. Preneel. Press Release: NESSIE Project Announces Final Selection of Crypto Algorithms, February 27, 2003. Available at `www.cryptonessie.org`.
18. B. Preneel, A. Biryukov, E. Oswald, B. Van Rompay, L. Granboulan, E. Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, M. Schafheutle, P. Serf, E. Biham, E. Barkan, O. Dunkelman, J.-J. Quisquater, M. Ciet, F. Sica, L. Knudsen, M. Parker, and H. Raddum. Nessie security report, version 2.0. Technical report, NESSIE Consortium, February 19 2003. Available at `www.cryptonessie.org`.
19. B. Schneier. *Applied Cryptography*. John Wiley & Sons Inc., New York, USA, 2nd edition, 1996.
20. Claude Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.
21. US Department of Commerce/National Institute of Standard and Technology. *FIPS PUB 197, Specification for the Advanced Encryption Standard (AES)*, November 2001. Available at `csrc.nist.gov/encryption/aes`.
22. G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the American Institute of Electrical Engineers*, XLV:109–115, 1926.
23. Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2003, 5th International Workshop, Cologne, Germany, September 8–10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*. Springer, 2003.
24. Thomas Wollinger, Sandeep Kumar. Fundamentals of Asymmetric Cryptography. This book.

Fundamentals of Asymmetric Cryptography

Thomas Wollinger and Sandeep Kumar

Horst Görtz Institute (HGI) for Security in Information Technology,
Ruhr University of Bochum, Germany
{wollinger, sandeep}@crypto.rub.de

Summary. Cryptology provides two different flavors of algorithms, namely symmetric and asymmetric (public–key) algorithms. This contribution deals with asymmetric algorithms.

Asymmetric cryptography provides the ability and is used in practical applications to: (a) exchange keys securely over a unprotected channel and (b) sign electronic document (Digital signature). Especially the first scenario is important in any kind of communication between systems. Hence, these cryptographic primitives are a necessity for securely exchanging messages in the car (e.g. between components) and between the car and a third party (e.g. tool station, other car, service provider).

This chapter gives first an introduction to asymmetric cryptography, helping the reader to understand the advantages as well as the problems. In the main part of the chapter we focus on two asymmetric cryptosystems, namely RSA and Elliptic Curve Cryptosystems (ECC). ECC is especially interesting for the usage in the automotive environment, because it is much better suited for the implementation on embedded processors. For each of the two cryptographic primitives we cover briefly the mathematical background and focus then on the engineering aspects (including fast implementation techniques) of these systems. In order to give the reader an idea about the performance of these algorithms we summarize available publications.

Keywords: asymmetric cryptography, embedded systems, cryptographical applications, efficient implementation, previous implementation

1 Introduction

Security services are provided by using cryptographic algorithms. There are two major classes of algorithms in cryptography: private–key or symmetric algorithms and public–key algorithms or asymmetric algorithms.

The main function of symmetric algorithms is the encryption of data, often at high speeds. However, symmetric–key cryptography has two main properties, namely (a) the algorithm requires the same secret key for encryption and decryption and (b) encryption and decryption algorithms are essentially

identical. Symmetric–key schemes are analogous to a safe box with a strong lock. Everyone with the key can deposit messages in it and retrieve messages. However, there are problems with symmetric–key schemes:

a. It requires secure transmission of a secret key, before being able to exchange messages.
b. In a network environment, each pair of users has to have a different keys resulting in too many keys ($\frac{n \cdot (n-1)}{2}$ key pairs). Hence, this fact results in problems handling the key management, the secure storage and so on.

Public–key (PK) cryptography introduces a new concept. The idea can be visualized, by making a slot in the safe box so that everyone can deposit a message (like a letter box). However, only the receiver can open the safe and look at its contents. This concept was proposed by Diffie and Hellman [13].

Section 2 concentrates on specific issues of public–key algorithms. The following two sections, Sections 3 and 4, concentrate on two specific public–key algorithms, namely RSA and ECC. In these sections we show the mathematical background, the computational aspects, and some implementation numbers. This contribution ends with our conclusions.

2 Public–Key Cryptography

Public–key cryptography is based on the idea of separating the key used to encrypt a message from the one used to decrypt it. Anyone who wants to send a message to a party, e.g., *Bob*, can encrypt that message using *Bob*'s *public key* but only *Bob* can decrypt the message using his *private key*. It is understood that the private key should be kept secret at all times and the public key is publicly available to everyone. Furthermore, it is impossible for anyone, except *Bob*, to derive the private key (or at least to do so in any reasonable amount of time). The basic protocol between the two communication parties Alice and Bob can be seen in Fig. 1, where K_{pubB} denotes the public key of Bob and K_{prB} the private (not publicly available) key of Bob.

One can realize three basic mechanisms with public–key algorithms: (a) Key establishment protocols (e.g., Diffi–Hellman key exchange) and key transport protocols (e.g., via RSA) without prior exchange of a joint secret, (b) Digital signature algorithms (e.g., RSA, DSA or ECDSA), and (c) Encryption. It looks as though public–key schemes can provide all functionality needed in modern security protocols such as SSL/TLS. However, PK systems have a major disadvantage when compared to private–key schemes. As stated above, public–key algorithms are very arithmetic intensive and – if not properly implemented or if the underlying processor has a poor integer arithmetic performance – this can lead to a poor system performance. Even when properly implemented, all PK schemes proposed to date are several orders of magnitude slower than the best-known private–key schemes. Hence, in practice,

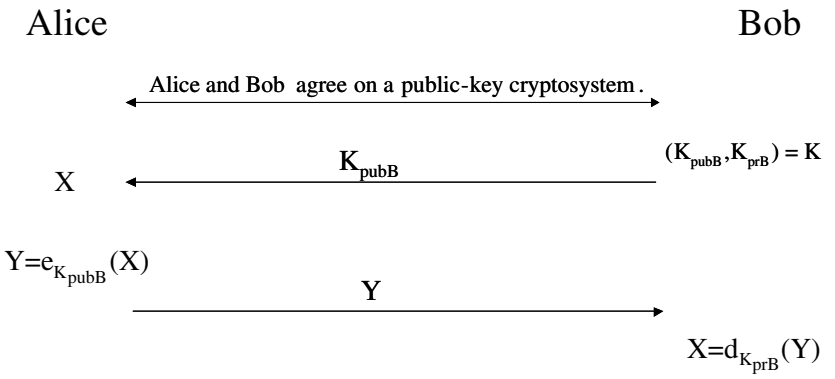

Fig. 1. Public–key encryption protocol

cryptographic systems are a mixture of symmetric–key and public–key cryptosystems and are call *hybrid cryptosystems*. Usually, a public–key algorithm is chosen for key establishment and then a symmetric–key algorithm is chosen to encrypt the communications, achieving in this way high throughput rates. Figure 2 summarizes the protocol steps from the exchange to the encryption of the data.

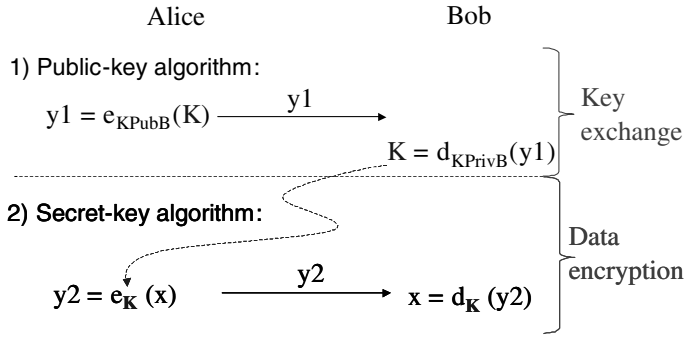

Fig. 2. Hybrid cryptosystem

Public–key algorithm are not only used for the exchange of a key, but also for the authentication by using digital signatures. Digital signatures are analogous to handwritten signatures. They enable communication parties to prove that one party has actually generated the message, also called `non-repudiation`.

The idea of the digital signature is appending a digital data block (like conventional signatures) to the message. Only the person who sends the message

must be capable of generating a valid signature (like conventional signatures). The signature is a function of a private key, so that only the holder of the private key can sign a message. For security reasons we have to change the signature with each document. In order to provide this functionality we make the signature a function of the message that is being signed. In practical terms we use the private key for signing (only Alice can sign her document using the KprA) and the public key for the verification (everyone can verify the signature, because KpuA is publicly known).

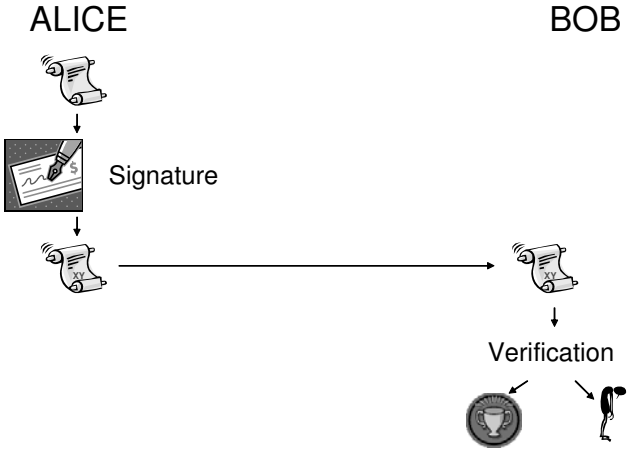

Fig. 3. Digital signature

Digital Signatures allow the following security services:

- Sender Authentication: Bob is sure that Alice signed the message, because only Alice knows her private key.
- Integrity: Message cannot be altered since that would be detected through verification.
- Non-repudiation: The receiver of the message can prove that the sender had actually sent the message. It is important to note that sender non-repudiation can only be achieved with public–key cryptography.

In general, one can divide practical public–key algorithms into three families:

- Algorithms based on the *integer factorization problem:* given a positive integer n, find its prime factorization, e.g. RSA [43].
- Algorithms based on the *discrete logarithm problem (DLP):* given α and β find x such that $\beta = \alpha^x \bmod p$, including the Diffie–Hellman key exchange protocol and the Digital Signature Algorithm (DSA).

- Algorithms based on *Elliptic Curves.* Elliptic curve cryptosystems [32, 24] are the most recent family of practical public–key algorithms which have gained acceptance including standardization [39].

In addition, there are many other public–key schemes, such as NTRU or systems based on hidden field equations, which are not in wide spread use. The scientific community is only at the very beginning of understanding the security of such algorithms.

Despite the differences between these mathematical problems, all three algorithm families have something in common: they all perform complex operations on very large numbers, typically 1024–2048 bits in length for the RSA and discrete logarithm systems or 160–256 bits in length for the elliptic curve systems.[1]. Table 1 puts the public–key bit length in perspective to the symmetric key algorithms. PK systems have a major disadvantage: they are very arithmetic intensive and even when properly implemented all PK schemes proposed to date are several orders of magnitude slower than the best-known private–key schemes.

Table 1. Key length for public–key and symmetric–key cryptography

Symmetric–key	ECC	RSA/DLP	
64 bit	128 bit	700 bit	only short term security (breakable with some effort)
80 bit	160 bit	1024 bit	medium term security (excl. government attacks)
128 bit	256 bit	2048-3072 bits	long term security (not assuming quantum computers)

Public–key schemes are based on modular exponentiation (RSA [43] and Discrete Logarithm (DL) based systems [13, 35]), i.e. the operation $x^e \bmod n$, or point multiplication (Elliptic Curve Cryptosystems [32, 24, 29, 35]), i.e. $k \times P$, where k is an integer and P is a point on the elliptic curve. Both operations are in their most basic forms performed via the right-to-left binary exponentiation algorithm [30] or one of its variants [16]. Performing such an exponentiation with, e.g., 1024-bit long operands is extremely computationally intensive. Interestingly enough, modular exponentiation with long numbers requires arithmetic which is similar to that performed in signal processing applications [8], namely integer multiplication.

The atomic operation in the right-to-left binary exponentiation algorithm is either modular multiplication, in the case of RSA and DL-based systems, or point addition, in the case of ECC, which in turn is performed through a combination of multiplications and additions on the field of definition of the elliptic curve.

[1] We refer to [26] for extended discussions regarding key equivalences between different asymmetric and symmetric cryptosystems.

Note that implementation of cryptographic systems presents several other requirements and challenges. As mentioned above, the performance of the algorithms is often crucial. One needs encryption algorithms to run at the transmission rates of the communication links. Slow–running cryptographic algorithms translate into consumer dissatisfaction and inconvenience. On the other hand, fast–running encryption might mean high product costs since higher speeds can be achieved through custom hardware devices and/or more code lines.

3 RSA

RSA is the most popular public–key algorithm and named for its creators – Rivest, Shamir, and Adelman [43]. In this section, we are first going to introduce the algorithm and then we show some techniques for efficient implementation as well as results of previous implementations. Considering the public–key algorithms it is also the easiest to understand and implement. RSA was patented until 2000 and is today free for use. RSA can be used for encryption and, thus, for key transport and digital signature. In the following we introduce the RSA algorithm consisting of: (a) set-up stage to produce the necessary public and private parameters needed and (b) the encryption/decryption function.

Set-up Stage

a. Choose two large primes p and q.
b. Compute $n = p \cdot q$.
c. Compute $\Phi(n) = (p-1)(q-1)$.
d. Choose random B; $0 < B < \Phi(n)$, with $\gcd(B, \Phi(n)) = 1$.
   Note that B has inverse in $Z_{\Phi(n)}$.
e. Compute inverse $A = B^{-1} \bmod \Phi(n)$: $B \cdot A \equiv 1 \bmod \Phi(n)$.
f. Public key: $k_{pub} = (n, B)$.
   Private key: $k_{pr} = (p, q, A)$.

Encryption: done using public key, k_{pub}.

$y = e_{k_{pub}}(x) = x^B \bmod n.$
$x \in Z_n = \{0, 1, \ldots, n-1\}.$

Decryption: done using private key, k_{pr}.

$x = d_{k_{pr}}(y) = y^A \bmod n.$

RSA has withstood all cryptanalyses since its introduction. The security is based on the difficulty of factoring large numbers. As shown above, the keys are functions of the product of two large primes p and q. If an attacker wants to find the plain text from a given cipher text, this is conjectured to be as difficult as factoring n. Hence, one should use at least 1024–bit long modulus n and private key a. For long–term security we strongly recommend at least 2048–bit.

3.1 Computational Aspects of RSA Encryption

The computation of the RSA encryption and decryption is dominated by modular exponentiation (e.g. encryption: $e_{k_{pub}}(x) = x^B \mod n = y$). In a naive way one could multiply x B-time to itself: $x \cdot x \cdot x \cdots x$. However, this implies that for one RSA operation one needs to compute $2^{1024} \approx 10^{300}$ multiplication, whereas B is a 1024 bit value. The elegant way to provide the modular exponentiation is to use the *square-and-multiply algorithm* also known as the *right-to-left binary exponentiation algorithm*. In the following we are going to introduce the most basic form of this algorithm; however, the authors in [30, 16] present more efficient variants of this algorithm based on the same concept. The most basic algorithm is reprinted in Algorithm 1. Average com-

Algorithm 1 Right-to-left binary exponentiation

Require: $B = \sum_{i=0}^{l-1} b_i 2^i$, x
Ensure: $z = x^B \mod n$
1: $z = x$
2: **for** $i = l - 1, 0$ **do**
3: $\quad z = z^2 \mod n$
4: $\quad$ **if** $b_i = 1$ **then**
5: $\quad\quad z = z \cdot x \mod n$
6: $\quad$ **end if**
7: **end for**

plexity of the right-to-left binary algorithms for an exponent B, whereas SQ and MUL denote field squaring operation and field multiplication operation, respectively:

$$[\log_2 l] \cdot \mathrm{SQ} + [\frac{1}{2} \log_2 l] \cdot \mathrm{MUL}$$

From Algorithm 1 one can see that the computationally most intensive operation for RSA is the modular multiplication. These operations have to be performed using very long operands. The operands of the RSA encryption algorithm should be at least 1024 bits long. Current desktop computers and embedded microcontrollers have word lengths of 8–64 bits. Hence, multiprecision arithmetic algorithms are required.

Besides the multiplication it is important to be able to have an efficient algorithm for modulo reduction. The straightforward way to apply modulo reduction uses division. The modular reduction needs to solve the following equation: $x \equiv r \bmod m$, whereas x, m is given. However, this integer division is very costly and we need a way to compute $x \bmod m$ without general division.

The problem of modular reduction has been extensively studied. Among the algorithms that have been proposed we find: Sedlak's Modular Reduction [48], Barret's Modular Reduction [2], Brickell's Modular Reduction [15], Quisquater's Modular Reduction, [40, 41, 10], and Montgomery's Modular Multiplication [34].

The Montgomery algorithm is a technique that allows efficient implementation of the modular multiplication without explicitly carrying out the modular reduction step and is the most widely used algorithm in the literature. However, the Montgomery multiplication is only beneficial if we compute a series of multiplications as we do when calculating the modular exponentiation. The algorithm to combine the Montgomery technique and the scalar multiplication can be found in [30].

The reader is referred to [30] for a list of all the different algorithms that can be used to realize the above–mentioned functionality.

3.2 Implementation Aspects of RSA

In this subsection we first introduce some issues that are of great use for practical applications. This is followed by implementation numbers published in the open literature. These numbers give the reader a rough idea about the performance of RSA in software and hardware.

For practical implementations the public exponent B is often chosen to be a short integer, for instance, the value $B = 17$ is popular. This makes encryption of a message (and verification of an RSA signature) a very fast operation. The reason for this is that one does not have to multiply the x to itself so many times (see Algorithm 1). However, the private exponent A needs to have full length, i.e., the same length as the modulus n, for security reasons. Note that a short exponent B does not cause A to be short.

The security of the implementation of cryptographic algorithms relies not only on mathematical functions. In the real world these cryptographic primitives are implemented on devices like computers and smart cards. In this cases these algorithms can behave quite differently from abstract mathematical functions. In the following we sketch some of the problems related to RSA implementation, namely secure padding, side channel attacks and fault injection.

In most of the practical applications it is necessary to apply padding for RSA. Encryption padding is necessary to avoid dictionary attacks. The padding adds a random string to the encrypted message and therefore it is not possible to build up dictionaries.

A common practice for signing with RSA is to apply a hash (or a redundancy) function to the message x, add some padding and raise the padded message to the decryption exponent. This is the basis of numerous standards such as ISO/IEC-9796-1 [23], ISO 9796-2 [22] and PKCS #1 v2.0 [25]. A good overview of the methods and difficulties of designing padding schemes can be found in [33] and current information can be found in [44].

Implementing cryptographic primitives one has to take caution of the physical side effects which are not considered in the traditional cryptographic model. Taking, for example, a smart card for the encryption of a message, this card requires an external power source and the operations performed on the smart card affect this power source. An adversary can monitor the power consumed and gains some information. This information can lead the adversary to the secret key on the smart card.

Power consumption is only one of the pieces of possible side channel information that the adversary can use. We can think of many more side channels, like current, time, radiation, temperature. A general overview of side channel attacks can be found in [46]. For specific side channel attacks and the necessary countermeasures for RSA implementations we refer the reader to [31] as well as to a survey of more recent research work [50].

Fault injection was first introduced in [6]. The authors showed how to break public–key algorithms, such as the RSA and Rabin signature schemes, by exploiting hardware faults. Subsequent publications introduced more sophisticated methods [3] for secret key algorithms. Meanwhile there have been many publications that show different techniques to insert faults, e.g., electromagnetic radiation [42], infrared laser [1], or even a flash light [49]. In order to avoid this kind of attack, we advise the development engineers to use *only* certified micro controllers or chip cards, the reason being, that standard chips are not protected.

To prevent this kind of attack, one needs to produce tamper–resistant devices. These devices will not only prevent fault injection, but also will stop an attacker from opening the chip and reading the information directly. Hence, this attack targets parts of the chip which are not available through the normal I/O pins. This can potentially be achieved through visual inspection and by using tools such as optical microscopes and mechanical probes. However, for more complex chips one can only launch such an attack with advanced methods, such as Focused Ion Beam (FIB) systems.

The interested reader is referred to the following citation for further instruction on how to construct a tamper–resistant device which prevents this kind of attack [53].

RSA in Software

In this subsection, we list some implementation numbers of RSA, in order to give the reader a feeling for the performance of this cryptographic primitive.

In [2], the Barret modular reduction method is introduced. The author implemented RSA on the TI TMS32010 DSP. A 512-bit RSA exponentiation took on average 2.6 seconds running at the DSP's maximum speed of 20 MHz. Reference [14] describes the implementation of a cryptographic library designed for the Motorola DSP56000 which was clocked at 20 MHz. The authors focused on the integration of modular reduction and multi-precision multiplication according to Montgomery's method [8, 34]. This RSA implementation achieved a data rate of 11.6 kbits/s for a 512-bit exponentiation using the Chinese Remainder Theorem (CRT) and 4.6 kbits/s without using it.

Reference [12] describes a fast software implementation of RSA, DSA, and ECDSA on a Pentium Pro@200 MHz running Windows NT 4.0 and using MSVC 4.2 and maximal optimization. The authors could achieve the RSA signature in 43.3 ms and the verification took 0.65 ms.
The fastest timings for 1024-bit RSA using CRT are 45.8 ms and 43567.8 ms for encryption and decryption, respectively. Signing took 43529.8 ms and the verification of a signature 213.6 ms.

Considering the above–mentioned numbers one could draw the conclusion that strong public–key cryptography is too computationally expensive for small devices. The authors in [19] implemented RSA-1024 and RSA-2048 on two 8-bit microcontrollers. To accelerate multiple-precision multiplication, they propose a new algorithm to reduce the number of memory accesses. Using a short exponent $e = 2^{16} + 1$, they attained a performance of 0.43 s for a RSA-1024 and 1.94 s for a RSA-2048 using an Atmel ATmega128 at 8 MHz.

RSA in Hardware

One of the few implementations of RSA on FPGA is presented in [5]. The authors implemented a version of Montgomery's algorithm optimized for a radix two hardware implementation. The authors used the Xilinx XC40250XV and the XC40150XV. The authors where able to run RSA-1024 encryption as fast as 0.22 ms and RSA-1024 decryption at 3.10 ms.

The latest implementation of RSA on an FPAG was presented in [28]. The authors introduced a serialization factor S. In the implementation, each instruction is broken up into several parts and executed in a serial fashion on S-bit operands. The implementation in [28] was targeted to a Xilinx Virtex-E 2000-8bg560. The implementation results of this contribution are summarized in Table 2.

In addition one can find some reported performance numbers on ASICs. In contrast to the above stated numbers these timings are from commercial products. SafeNet Inc. produces the SafeXcel 1842 chip [45]. SafeXcel 1842 is able to compute the RSA 1024-bit signatures in 476 μs. The Cavium chip CN1540 NitroxPlus is able to compute the same operation in 22 μs [7].

Table 2. RSA encryption with 1024 bit key [28]

S	T_{ck} [ns]	Area [slices]	Total Time [ms]
32	8.74	995	5.64
64	11.6	1188	3.86
128	18.6	1870	3.27
256	30.9	2902	2.99

Evaluation of RSA Implementations

RSA is the most widely used encryption and decryption algorithm in cryptographic application. The acceleration of RSA using dedicated hardware or coprocessors is commonly used. Software implementation on general–purpose processors [12] and targeting embedded processors [2, 14, 11, 19] reach only moderate performance due to the long operand length. Thus, hardware accelerators, e.g. using FPGAs or ASICs, are necessary, to atttain a major speed–up compared to software implementation. One notices, when consulting the given references, that the throughput is some orders of magnitude higher when using FPGA. The ASIC encryption speed can even outperform the FPGA performance. The performance of a FPGA implementation can be increased by using specific characteristics of the FPGA, e.g. compact implementation of shift registers via distributed RAM blocks or multiplexer via tristate buffer [28].

4 Elliptic Curve Cryptosystem (ECC)

Elliptic Curve Cryptosystem is a relatively new cryptosystem, suggested independently in 1987 by Koblitz [24] at the University of Washington and in 1986 by Miller at IBM [32]. ECC can be used instead of Diffie–Hellman and other DL-based algorithms. In this section we provide a brief introduction to elliptic curve cryptosystems covering also point addition and doubling. In the second part we give some techniques on how to implement ECC. In the third part of the section we introduce some performance number in software and hardware. Additional information can be found in [32, 24, 4].

Elliptic curve cryptosystem is based on the discrete logarithm problem (DLP). The basic operation that needs to be performed for ECC is $Q = k \cdot P$, where k is an integer and P is a point of a finite group. This operation is known as scalar multiplication. Hence, we have to add the point P k-times to itself to get the solution. Figure 4, illustrates the DLP. The finite group is represented by the cloud and the dots are the finite elements of the group. P is added k-times to itself until we get the necessary result Q. Thus, we hop from one point (element of the group) to another point until we are at point Q. The DLP is based on the assumption that an attacker knows P and Q,

but has to find k. Thus, the attacker needs to find out how many times we had to "jump" until we got from the starting element to the resulting element. Hence, given P and Q, it is a hard problem to obtain k.

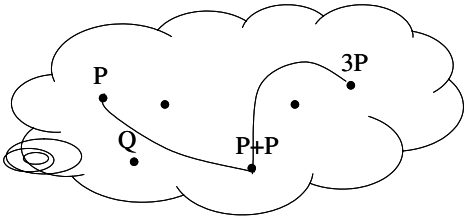

Fig. 4. Repeated addition in a finite group

An elliptic curve E over $GF(p)$ is the set of solutions $P = (x, y)$ which satisfy the Weierstrass equation:

$$E: \; y^2 = x^3 + Ax + B \pmod{p} \tag{1}$$

where $A, B \in GF(p)$ and $4A^3 + 27B^2 \not\equiv 0 \bmod M$ with $M > 3$, together with the point at infinity $\mathcal{O}$. Figure 5 shows an example of an elliptic curve displayed over the reals.

Following the discussion above, we now have to find a (cyclic) group $(\mathcal{G}, \circ)$ so that we can use the DL problem as a one-way function. The set of elements is given by the points on the curve. We "only" need a group operation on the points. Hence,

- **Group $\mathcal{G}$:** Points on the curve given by (x, y).
- **Operation $\circ$:** $P + Q = (x_1, y_1) + (x_2, y_2) = R = (x_3, y_3)$.

Finding $P + Q = R$ in a geometrical manner can be achieved by the following two steps and is shown in Figure 5:

a. $P \neq Q \rightarrow$ line through P and Q and mirror point of third interception along the x-axis.
b. $P = Q \Rightarrow P + Q = 2Q \rightarrow$ tangent line through Q and mirror point of second intersection along the x-axis.

The way to implement the group operation (the addition of points on the elliptic curve) is described in the following section.

4.1 Computational Aspects of ECC

Elliptic curve cryptosystems depend on arithmetic involving the points of the curve. Hence, one needs to provide group addition and group doubling operation. The curve equation (Equation 1) gives us the points on the curve

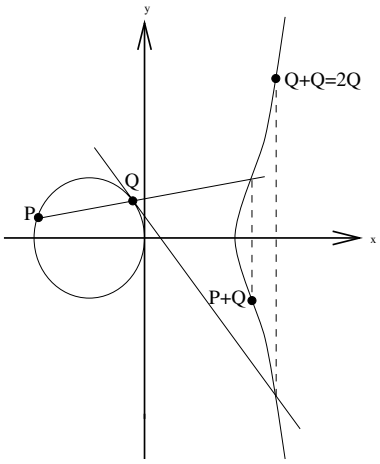

Fig. 5. $y^2 = x^3 + a \cdot x + b$ over the reals

and therefore the group elements to compute the scalar multiplication ($Q = k \cdot P$). The curve arithmetic is defined in terms of underlying field operation. Hence, the efficiency of the field operation and group operation is likewise crucial to the overall performance.

ECC based on GF(p)

It is well known that the points on elliptic curve form a group under an addition operation which is defined as follows. Let $P = (x_0, y_0) \in E$; then $-P = (x_0, -y_0)$. $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E$. If $Q = (x_1, y_1) \in E$ and $Q \neq -P$, then $P + Q = (x_2, y_2)$, where

$$x_2 = \lambda^2 - x_0 - x_1 \tag{2}$$

$$y_2 = \lambda(x_1 - x_2) - y_1 \tag{3}$$

and

$$\lambda = \begin{cases} \frac{y_0 - y_1}{x_0 - x_1} & \text{if } P \neq Q \\ \\ \frac{3x_1^2 + A}{2y_1} & \text{if } P = Q \end{cases} \tag{4}$$

To perform the above introduced group operation we need to provide the underlying field arithmetic. For now we only consider elliptic curves using the underlying prime field. Hence, we have to provide field operations based on prime fields. The crucial field operation is modular multiplication. Hence, we can use the techniques described for the RSA implementation. The difference is that the modulus is a prime number for ECC and in the case of RSA it is a product of two primes (therefore a composite number).

ECC based on GF(2^m)

Considering different application and platform characteristics two fields might be of interest. An elliptic curve E over $GF(2^m)$ is the set of solutions $P = (x, y)$ which satisfy the equation:

$$E: \; y^2 + xy = x^3 + Ax + B \pmod{p} \tag{5}$$

The group law will change accordingly: $P + Q = (x_2, y_2)$, where $x_2 = \lambda^2 + \lambda + x_0 + x_1 + A$, $y_2 = \lambda(x_0 + x_2) + x_1 + y_0$. and

$$\lambda = \begin{cases} \frac{y_0 - y_1}{x_0 - x_1} \text{ if } P \neq Q \\ \\ \frac{x_0^2 + y_0}{x_0} \text{ if } P = Q \end{cases} \tag{6}$$

Again the crucial operation is modular multiplication. The next paragraphs will describe how to perform reduction and multiplication considering the characteristic two fields efficiently. The modulo multiplication considering the characteristic two fields can be implemented efficiently in two separate steps, in contrast to the Montgomery multiplication (including reduction) used in the odd characteristic case.

Different Coordinate Representations

In addition to the different underlying fields one can use varieties of coordinate representation. The coordinate representation considered so far is known as affine representation. However, in many applications it is more convenient to represent the points P and Q in projective coordinates. This is advantageous when inversion is computationally expensive compared to multiplication in the finite field. Thus, algorithms for projective coordinates trade inversions in the point addition and in the point double operations for a larger number of multiplications and a single inversion at the end of the algorithm. This inversion can be computed via exponentiation using the fact that $A^{-1} \bmod M \equiv A^{M-2} \bmod M$. In projective coordinates, a point $P = (x, y)$ is represented as $P = (X, Y, Z)$ where $X = x$, $Y = y$, and $Z = 1$. To convert from projective coordinates back to the affine ones, we use the following relations:

$$x = \frac{X}{Z^2}, \;\; y = \frac{Y}{Z^3}$$

Finally, one can obtain expressions equivalent to (4) and (6) for doubling and addition operations in projective coordinates. We refer to [21] for the actual algorithms. In Tables 3 and 4 we present the complexity of the group operations considering different coordinate representations. Note that in this context the complexity of adding or doubling a point on an elliptic curve is usually given by the number of field multiplications and inversions (if affine coordinates are being used). Field additions are relatively cheap operations compared to multiplications or inversions and therefore are neglected in the tables.

Table 3. Operation counts for point addition and doubling on $y^2 = x^3 - 3x + B$. A = affine, P = standard projective, J = Jacobin, C = Chudnovsky, I = field inversion, M = field multiplication, S = field squaring [21]

Doubling		General addition		Mixed coordinates	
$2A \to A$	$1I, 2M, 2S$	$A + A \to A$	$1I, 2M, 1S$	$J + A \to J$	$8M, 3S$
$2P \to P$	$7M, 3S$	$P + P \to P$	$12M, 2S$	$J + C \to J$	$11M, 3S$
$2J \to J$	$4M, 4S$	$J + J \to J$	$12M, 4S$	$C + A \to C$	$8M, 3S$
$2C \to C$	$5M, 4S$	$C + C \to C$	$11M, 3S$		

Table 4. Operation counts for point addition and doubling on $y^2 + xy = x^3 - Ax^2 + B$. M = field multiplication, D = field division [21]

Coordinate system	General addition	Mixed coordinates	Doubling
Affine	D + M	–	V + M
Standard projective	13M	12M	7M
Jacobian projective	14M	8M	4M
López-Dahab projective	14M	8M	4M

4.2 Implementation Aspects of ECC

In this subsection we first list some facts that are helpful for practical applications. This part is followed by some implementation numbers published in software and hardware.

The patent issue for elliptic curve cryptosystems is the opposite of that for RSA and Diffie–Hellman, where the cryptosystems themselves have patents. Elliptic curve cryptosystems have no general patents, though some newer elliptic curve algorithms and certain efficient implementation techniques may be covered by patents.

For example, Apple Computer holds a patent on efficient implementation of odd-characteristic elliptic curves, including elliptic curves over GF(p) where p is close to a power of 2. Certicom holds a patent on efficient finite field multiplication in normal basis representation, which applies to elliptic curves with such a representation and also Cylink holds a patent on multiplication in normal basis. In all of these cases, it is the implementation technique that is patented, not the prime or representation, and there are alternative, compatible implementation techniques that are not covered by the patents.

In almost all cases, when an adversary broke a system, he did find some security lacks in the implementation. Hence, he did not break the mathematical functions, but rather found some problems with the software/hardware using the cryptographic primitive. For the implementation of ECC it is important to note that there are many side channel attacks and that the security engineer should also protect against fault injection.

Any implementation will leak side channel information such as power, time, radiation or temperature. The aim of the implementation is to keep this

information as small as possible so that it is *not* possible for an unauthorized party to guess the secret. For a general introduction to side channel attacks we refer the interested reader to [50].

In the case of ECC we find many publications dealing with possible side channel attacks and their counter–measures. A good number of the side channel attacks can be prevented on the algorithmic level, the so–called algorithmic counter–measures.

Fault injection is an attack that exploits hardware faults. This fault can be caused for example through electromagnetic radiation or infrared laser. An attacker has to place the faults at the right position on a chip and/or at an appropriate time to get the secret information out of the cryptographic device. The solution for preventing this kind of attack is tamper–resistant chips. We refer the reader to the following citations for information about the techniques for tamper–resistant designs [53]. A tamper–resistant application also does not allow physical inspection and therefore it is not possible for the attacker to visual inspect the chip.

ECC in Software

This subsection gives a brief overview of some implementation numbers published. The Aim of the section is to give the reader a feeling for the performance of ECC in software using different processors.

Table 5 lists some ECC performance numbers for embedded and general processors as targeting platform. Generally speaking we can perform the ECC scalar multiplication using embedded platforms in some seconds, whereas general–purpose computers will need only some seconds. The reason is the limited resources of the first type of processors.

ECC in Hardware

The performance of ECC over binary fields on FPGAs has been extensively studied in recent years. In the following paragraphs describe in more detail the latest and fastest ECC implementations on FPGAs.

In [37], the authors introduced a processor architecture for elliptic curve cryptosystems over binary fields. The architecture is scalable in terms of area and speed and therefore can be optimized for different curves and fields. The processor consists of three main components: main controller, the arithmetic unit controller, and the arithmetic unit. The main controller interacts with the host system and controls the computation of kP. The arithmetic unit controller is responsible for the group operations, the coordinate conversions and controls the arithmetic unit (AU). The AU performs the field operations, like addition, squaring, multiplication and inversion. The most important field arithmetic components are the optimized bit-parallel squarer and a digit-serial multiplier, which can be reconfigured for different field orders and field polynomials. Using projective coordinates and the Montgomery scalar multiplication

Table 5. Timings for ECC scalar multiplications on different platforms

Platform		field and multiplication method	t_m [ms]
Embedded processor	Siemens SLE44C24S @5MHz [52]	$GF((2^8-17)^{17}) \approx GF(2^{134})$	
		binary-double-and-add	8370
		de Rooij w/18 precomputations	1830
	TI MSP430x33x @1MHz [17]	$GF(2^{128}-2^{97}-1) \approx GF(2^{128})$	
		binary-double-and-add (asm)	3400
	PDA Handspring Visor @16MHz [51] (Motorola Dragonball)	$GF(2^{163})$	
		Koblitz: $\tau - adic$	1670
		Koblitz: $\tau - adic$ width-4	1510
		Koblitz: w/18 precomputations	870
	ATmega128 @8MHz [19]	SECG-$GF(2^{224})$	2190
		SECG-$GF(2^{192})$	1240
		SECG-$GF(2^{160})$	810
General–purpose processor	Sun UltraSPARC @300MHz [27]	$GF(2^{163})$	
		Montgomery w/o precomp.	10.50
	DEC Alpha 64-bit @175MHz [47]	$GF(2^{155})$	
		almost inv.	7.80
	Sun Fire 280R @900MHz [18]	$GF(2^{163})$	3.11
	PentiumII @400MHz [20]	$GF(2^{163})$	1.68
	PentiumII @200MHz [12]	$GF(2^{191})$	0.50

algorithm, they could calculate a scalar multiplication with arbitrary points in 0.21 ms considering the underlying field GF(2^{167}). At the time this result was presented, it was the fastest ECC implementation on an FPGA.

In [18], the authors presented a programmable hardware accelerator to speed up point multiplication for elliptic curves of binary fields. The introduced architecture is scalable and handles curves with arbitrary fields up to $m = 255$. The authors also hardwired some of the commonly used curves to obtain higher performance. The multiplier took 73% of the look-up table (LUT) and 46% of the flip-flops. In addition, the multiplication took 62% of the execution time and thus the large portion of resources used was justified. They could gain better performance by using parallel execution and by separating control flow and data flow. Considering the hardwired curves over GF(2^{163}) they could perform a point multiplication in 0.14 ms, which is to our knowledge the fastest result in the open literature.

In addition, there are very few implementations in the open literature using prime fields. In [38], the authors proposed an elliptic curve processor (ECP) architecture for the computation of point multiplication for curves defined over fields $GF(p)$. The targeted platform was a Xilinx XCV1000E-8-BG680 (Virtex E) FPGA. This prototype was programmed to support the

fields $GF(2^{192} - 2^{64} - 1)$ and $GF(2^{521} - 1)$. The design for $GF(2^{192} - 2^{64} - 1)$ used 11,416 LUTs, 5735 flip-flops, and 35 BlockRAMS. The frequency of operation of the prototype was 40 MHz for 192–bit operands and 37.3 MHz for the 521-bit multiplier. The authors in [38] estimated the time to compute a point multiplication for an arbitrary point on a curve defined over $GF(2^{192} - 2^{64} - 1)$ as approximately 3 ms.

ASIC implementation reaches a similar speed to the fastest FPGA implementation. The CE71 targeting 0.25 μm and using 165k gates could perform an ECC scalar multiplication in 1.10 ms. This speed was achieved using a random curve based on $GF(2^{163})$ [36]. The Koblitz curve can be performed on the same chip in 0.65 ms.

Evaluation of ECC Implementations

ECC based on GF(2^m) fields are well suited for hardware implementations, the reason being that the representation of the elements maps the hardware structure. FPGA and ASIC implementations are some factors faster compared to software implementations and can therefore be used as accelerators when additional speed is necessary.

GF(p) and GF(p^m) seem to be better suited for general–purpose computers, since they can utilize fast multiplication and division instructions. Furthermore, GF(p^m) is the better choice, because no precision routines are needed [9].

5 Conclusions

After a short introduction to public–key (asymmetric) cryptography, we concentrated on the engineering aspects of the primitives RSA and ECC. RSA is the most widely used asymmetric algorithm and ECC is especially promising for embedded application, due to the short key length compared to RSA. We summarized the mathematical background of these algorithms and introduced the state-of-the-art techniques necessary to implement them efficiently. We also give the reader an insight into the best published performance numbers of these algorithms implemented on different platforms, namely embedded microprocessors, general processors, FPGAs, and ASICs.

References

1. C. Ajluni. Two New Imaging Techniques to Improve IC Defect Indentification. *Electronic Design*, 43(14):37–38, July 1995.
2. P. Barrett. Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor. In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, LNCS 263, pages 311–323, Berlin, Germany, August 1986. Springer-Verlag.

3. E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In Burt Kaliski, editor, *Advances in Cryptology – Crypto '97*, pages 513–525, Berlin, 1997. Springer-Verlag. LNCS 1294.
4. I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, London Mathematical Society Lecture Notes Series 265, 1999.
5. T. Blum and C. Paar. Montgomery modular multiplication on reconfigurable hardware. In *Proceedings of the 14th IEEE Symposium on Computer Arithmetic (ARITH-14)*, pages 70–77, 1999.
6. D. Boneh, R. A. DeMillo, and R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In Walter Fumy, editor, *Advances in Cryptology – EuroCrypt '97*, pages 37–51, Berlin, 1997. Springer-Verlag. LNCS 1233.
7. Cavium. CN1540, NitroxPlus. `www.cavium.com`, 2004.
8. Ç. K. Koç, T. Acar, and B. Kaliski. Analyzing and Comparing Montgomery Multiplication Algorithms. *IEEE Micro*, pages 26–33, June 1996.
9. Jae Wook Chung, Sang Gyoo Sim, and Pil Joong Lee. Fast Implementation of Elliptic Curve Defined over $GF(p^m)$ on CalmRISC with MAC2424 Coprocessor. In Çetin K. Koç and Christof Paar, editors, *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2000*, LNCS 1965, pages 57–70, Berlin, 2000. Springer-Verlag.
10. D. De Waleffe and J. J. Quisquater. CORSAIR: A smart card for public key cryptosystems. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology – CRYPTO '90*, LNCS 537, pages 502–514, Berlin, 1990. Springer-Verlag.
11. Henna Pietiläinen. Elliptic curve cryptography on smart cards. Master's thesis, Helsinki University of Technology, October 2000.
12. E. DeWin, S. Mister, B. Preneel, and M. Wiener. On the Performance of Signature Schemes Based on Elliptic Curves. In J. P. Buhler, editor, *Algorithmic Number Theory: Third International Symposium (ANTS 3)*, LNCS 1423, pages 252–266. Springer-Verlag, June 21–25 1998.
13. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.
14. S. R. Dussé and B. S. Kaliski. A Cryptographic Library for the Motorola DSP56000. In I. B. Damgård, editor, *Advances in Cryptology – EUROCRYPT '90*, LNCS 473, pages 230–244, Berlin, Germany, May 1990. Springer-Verlag.
15. E. F. Brickell. A fast modular multiplication algorithm with applications to two key cryptography. In D. Chaum, R. L. Rivest and A. T. Sherman, editors, *Advances in Cryptology – CRYPTO '82*, pages 51–60, New York, USA, 1982. Plenum Publishing.
16. D. M. Gordon. A Survey of Fast Exponentiation Methods. *Journal of Algorithms*, 27:129–146, 1998.
17. J. Guajardo, R. Bluemel, U. Krieger, and C. Paar. Efficient Implementation of Elliptic Curve Cryptosystems on the TI MSP430x33x Family of Microcontrollers. In K. Kim, editor, *Fourth International Workshop on Practice and Theory in Public Key Cryptography – PKC 2001*, LNCS 1992, pages 365–382, Berlin, February 13-15 2001. Springer-Verlag.
18. N. Gura, S. Chang, H. Eberle, G. Sumit, V. Gupta, D. Finchelstein, E. Goupy, and D. Stebila. An End-to-End Systems Approach to Elliptic Curve Cryptography. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, LNCS 1965, pages 351–366. Springer-Verlag, 2001.

19. N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, LNCS. Springer-Verlag, 2004.
20. D. Hankerson, J. López Hernandez, and A. Menezes. Software Implementation of Elliptic Curve Cryptography Over Binary Fields. In Ç. Koç and C. Paar, editors, *Second International Workshop on Cryptographic Hardware and Embedded Systems – CHES 2000*, LNCS 1965, Berlin, 2000. Springer-Verlag.
21. D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, New York, USA, 2004.
22. ISO. *ISO/IEC 9796-2: Information technology – Security techniques – Digital signature scheme giving message recovery, Part 2: Mechanisms using a hash-function*, 1997.
23. ISO. *ISO/IEC 9796: Information technology – Security techniques – Digital signature scheme giving message recovery, Part 1: Mechanisms using redundancy*, 1999.
24. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.
25. RSA Laboratories. *PKCS #1: RSA cryptography specifcations, version 2.0,*, September 1998.
26. A. K. Lenstra and E. R. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14(4):255–293, 2001.
27. J. López and R. Dahab. Fast Multiplication on Elliptic Curves over $GF(2^n)$. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 1999*, LNCS 1717, pages 316–327. Springer-Verlag, 1999.
28. M. Mazzeo, L. Romano, G. P. Saggese, and M. Mazzocca. FPGA-based Implementation of a serial RSA processor. In *Design, Automation and Test in Europe Conference and Exhibition (DATE'03)*, pages 10582–10590, March 2003.
29. A. Menezes and D. Johnson. The elliptic curve digitial signature algorithm (ECDSA). Technical report CORR 99-34, Department of C & O, University of Waterloo, Ontario, Canada, August 1999.
30. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, USA, 1997.
31. T. S. Messerges, E. A. Dabbish, R. H. Sloan Power Analysis Attacks of Modular Exponentiation in Smartcards. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 1999*, LNCS 1717, pages 144–157, Berlin, 1999. Springer-Verlag.
32. V. Miller. Uses of Elliptic Curves in Cryptography. In H. C. Williams, editor, *Advances in Cryptology – CRYPTO '85*, LNCS 218, pages 417–426, Berlin, Germany, 1986. Springer-Verlag.
33. J. F. Misarsky. How (not) to design signature schemes. In Hideki Imai and Yuliang Zheng, editors, *First International Workshop on Practice and Theory in Public Key Cryptography – PKC'98*, LNCS 1431, pages 14–28, Berlin, 1998. Springer-Verlag.
34. P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, April 1985.
35. US Department of Commerce/National Institute of Standard and Technology. *Digital Signature Standard (DSS)*, January 27, 2000.
36. S. Okada, N. Torii, K. Itoh, and M. Takenaka. Implementation of Elliptic Curve Cryptographic Coprocessor over $GF(2^m)$ on an FPGA. In Çetin K. Koç and

Christof Paar, editors, *Proceedings of the Second Workshop on Cryptographic Hardware and Embedded Systems – CHES 2000*, pages 25–52, Berlin, Germany, 2000. Springer-Verlag.
37. G. Orlando and C. Paar. A High-Performance Reconfigurable Elliptic Curve Processor for $GF(2^m)$. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, LNCS 1965, pages 41–56. Springer-Verlag, 2000.
38. G. Orlando and C. Paar. A Scalable $GF(p)$ Elliptic Curve Processor Architecture for Programmable Hardware. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, LNCS 2162, pages 348–363. Springer-Verlag, 2001.
39. *IEEE P1363-2000: IEEE Standard Specifications for Public Key Cryptography*, 2000. Available at `standards.ieee.org/catalog/olis/busarch.html`.
40. J.-J. Quisquater. Fast modular exponentiation without division. Rump session of EUROCRYPT '90.
41. J.-J. Quisquater. Encoding system according to the so-called RSA method, by means of a microcontroller and arrangement implementing this system. United States Patent, Patent Number 5166978, November 24 1992.
42. J.-J. Quisquater and D. Samyde. Electro Magnetic Analysis (EMA): Measures and Countermeasures for Smart Cards. In *International Conference on Research in Smart Cards, E-smart 2001*, pages 200–210, Cannes, France, September 2001.
43. R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and public–key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
44. RSA Labratories. `www.rsasecurity.com/rsalabs`.
45. SafeNet. SafeXcel 1842. `www.safenet-inc.com`, 2004.
46. K. Schramm, K. Lemke, and C. Paar. Embedded Cryptography: Side Channel Attacks. This book.
47. R. Schroeppel, H. Orman, S. O'Malley, and O. Spatscheck. Fast key exchange with elliptic curve systems. In D. Coppersmith, editor, *Advances in Cryptology – CRYPTO '95*, LNCS 963, pages 43–56, Berlin, Germany, 1995. Springer-Verlag.
48. H. Sedlak. The RSA cryptography processor. In D. Chaum and W. L. Price, editors, *Advances in Cryptology – EUROCRYPT '87*, LNCS 304, pages 95–105, Berlin, Germany, 1987. Springer-Verlag.
49. S. Skorobogatov and R. Anderson. Optical Fault Induction Attacks. In Ç. K. Koç B. S. Kaliski and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, LNCS 2523, pages 2–12. Springer-Verlag, 2002.
50. The Side Channel Cryptanalysis Lounge. `www.crypto.rub.de/en_sclounge.html`.
51. A. Weimerskirch, C. Paar, and S. Chang Shantz. Elliptic Curve Cryptography on a Palm OS Device. In V. Varadharajan and Y. Mu, editors, *The 6th Australasian Conference on Information Security and Privacy – ACISP 2001*, LNCS 2119, pages 502–513, Berlin, 2001. Springer-Verlag.
52. A. Woodbury, D. V. Bailey, and C. Paar. Elliptic curve cryptography on smart cards without coprocessors. In *IFIP CARDIS 2000, Fourth Smart Card Research and Advanced Application Conference*, Bristol, UK, September, 2000. Kluwer.
53. Kerstin Lemke. Embedded Security: Physical Protection Against Tampering Attacks. This book.

Security Aspects of Mobile Communication Systems

Jan Pelzl and Thomas Wollinger

Horst Görtz Institute (HGI) for Security in Information Technology,
Ruhr University of Bochum, Germany
{pelzl,wollinger}@crypto.rub.de

Summary. Communication in modern applications increasingly moves to wireless systems. There exist manifold wireless communication protocols which might be integrated in future car applications. In this contribution, we introduce the most important standards for wireless communication and show associated security implications.

1 Introduction

In recent years, applications have increasingly moved from wired to wireless. Former localized applications (e.g. desktop computers, phones) are now wireless and, therefore, movable. The convenience and acceptance of wireless and agile products yield an increasing demand for similar products in cars. Additionally, car manufacturers can reduce manufacturing costs by substitution of wired applications. The massive amount of cables in cars adds to cost and weight and can be avoided with wireless applications. Hence, we will see not only wireless entertainment applications, but also car functions move to wireless connections.

Communication of data using the air as a channel is very vulnerable to attacks. The reason is that the attacker can easily tap the channel using radio receivers. The attacker is able to read, change, or delete messages. This can be a tremendous problem, leading to great financial losses for manufacturers even if the attack targets less important features in cars. The loss of confidence in car brands due to malfunctions in electronic devices is a concern of today's manufacturers. An attacker could invoke malfunctions of simple services such as power windows or air conditioning. The driver will probably return the car and never buy the same brand again.

Nowadays, several protocols for wireless communication can be found to be used for any new application that might be integrated in future cars. These include applications for in-car communication, car-to-car communication, and far field communication. However, we have to analyze these systems carefully

in order to understand their security limitations. In this chapter, we introduce the most important standards for wireless communication and show associated security implications. All wireless systems can be structured by the designated application focus:

- *Far field communication:* Cell phone networks such as the Global System for Mobile Communication (GSM) and the Universal Mobile Telecommunication System (UMTS). These systems can be used for several services such as traffic information, toll billing, weather information, local information services, and dynamic routing.
- *Car-to-car and hot-spot communication:* Wireless network standards wireless LAN (WLAN) and HiperLan/2 are appropriate for small-range data interchange between cars. Applications are, e.g., safety systems, ad-hoc networking between cars, remote diagnosis, and hot-spot communication (e.g., at gas stations).
- *In-car communication:* Possible applications of Bluetooth, ZigBee, DECT, and IrDA include data exchange from sensors to the control network of the car, PDA data interchange with car networks, and identification with keys.

For each standard we first give a short description of the architecture focusing on the part providing the security services. In addition, we list the security services as well as the security shortcomings of the standards.

Finally, a brief comparison of all standards regarding technical and security related aspects is given in Table 1.

2 Global System for Mobile Communication (GSM)

In the 1980s, most mobile cellular systems were based on analog technology. The Global System for Mobile communication can be considered as the first digital system. In 1982, the idea for an European standard for mobile communication over the band 900 MHz was born. By 1985 Germany, France, and Italy had signed an agreement for development of such a standard system. In 1991 the first GSM system was established in Genf. Today, GSM is a digital mobile telephone system that is widely used in Europe and other parts of the world.

In the car environment GSM can be used for far-field communication. Examples of an application could be remote diagnosis or on-board Internet. GSM can be used for transmitting the necessary data wirelessly to the manufacturer who analyzes the diagnosis data or for the exchange of data with the Internet provider. For the first application it is of great importance that the data arrives correctly to allow a complete diagnosis, whereas for the second application we want mainly to have a reliable system to guarantee consumer satisfaction. An example for current use of GSM in the automotive is "Toll Collect", a toll collecting system for German highways. However, security plays a central role in keeping the transmitted information private.

2.1 Overview

In this subsection we describe very briefly the main parts of the GSM network, concentrating on the part applying the security mechanisms.

Mobile Station (MS): The mobile station consists of the hardware (the cell phone) itself and the Subscriber Identity Module (SIM). The phone is uniquely characterized through the International Mobile Equipment Identity (IMEI) and provides the encryption algorithm A5. The A5 algorithm achieves encryption for the data transmission. The SIM is a smart card providing the user with access to the subscriber services. A four-digit PIN (Personal Identification Number) identifies the user to the chip card. In addition, the following information is stored on the card: IMSI (International Mobile Subscriber Identity), the user-specific symmetric Key K_i (128 bit), the A3 algorithm for challenge-and-response authentication and the A8 algorithm to generate the session key.

Base Station Subsystem (BSS): The BSS handles the connection between MS and the Network and Switching Subsystem (NSS). The BSS can be divided into two parts: (a) the Base Transceiver Station (BTS) and (b) the Base Station Controller (BSC).

Network and Switching Subsystem (NSS): The main aim of the NSS is to manage the communication between the different users, including the storage of information concerning the subscribers and the coordination of their mobility. NSS consists of the Mobile services Switching Center (MSC), the Gateway Mobile services Switching Center (GMSC), the Home Location Register (HLR), the Visitor Location Register (VLR), and the Authentication Center (AuC) Equipment Identity Register (EIR).

HLR, VLR, and AuC are the components of the NSS important for security. HLR stores subscriber-related data, hence, also the cryptographically related date, like user keys. The VLR is a temporary database for visitors needed to ensure services. The network provider needs to deposit the encryption data and authentication of the MS. AuC verifies the identity of the user by providing authentication and encryption parameters.

Operation and Support Subsystem (OSS): The OSS is connected to the NSS and to the BSC and allows for monitoring and controlling the system.

2.2 Security of the GSM network

In this section, we will provide the reader with an overview of the security aspects implemented in GSM. GSM provides two cryptographic protocols, namely to ensure authentication of the user to the network and to encrypt the data.

Authentication of the User to the Network: Authentication allows the network provider to uniquely identify the user by checking if the user (or better the chip card of the user) knows the IMSI and the user key K_i. Authentication avoids therefore the placing of calls to another person's account.

Figure 1 illustrates the authentication between user and network. In the first step the user sends the TIMSI (Temporary IMSI) to the base station. The idea behind the TIMSI is to hide the identity of the MS by frequently updating this number during each location update procedure. The TIMSI allows the network to get the user's key K_i from the database. In the second step, the network challenges the SIM card with a 128-bit random number. At this point the two parties (SIM and network) are able to calculate the Signed Response (SRES) using the A3 algorithm. The cell phone is authenticated if the two results are equal.

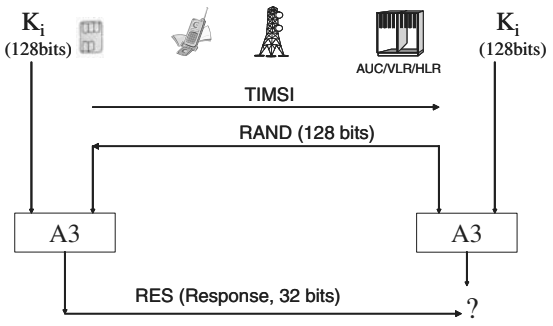

Fig. 1. GSM user authentication

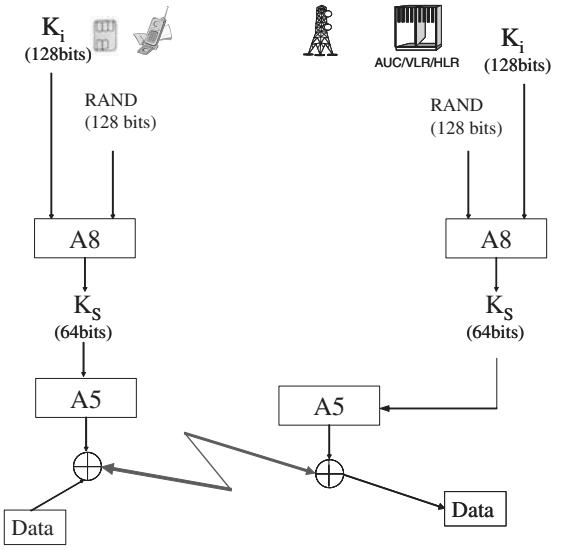

Fig. 2. GSM voice encryption

Voice Encryption: Data is transferred by air and, thus, can be easily eavesdropped by an attacker using radio receivers. Encryption is done using a session key K_S which is generated using the A8 algorithm (see Figure 2). The input values to A8 is a random number RAND and the user key K_i. Note that K_S is valid at most for one communication. The cipher A5 uses K_S as input and produces the key stream to be XORed with the data.

2.3 Security Analysis

In this subsection we analyze the security features of GSM. It is important to understand the strengths but also the drawbacks of GSM. Using this knowledge an application designer is able to judge whether the system is sufficient in terms of security for the given application. For more detailed information the reader is referred to [13, 17].

Access control: Access control between the user and the SIM card is given by a secret PIN and, thus, does not allow (easy) unauthorized access. Note that the PIN is not a sufficient protection against serious attacks.

Downlink signalling failure counter: When an mobile subscriber enters the influence area of a jamming unit, it loses contact with the active carrier of its BS. After the downlink signaling failure counter (DSC) expires after a certain time, the MS considers a failure has taken place.

Temporary identity: The TIMSI will be newly assigned to the user at each location. Hence, it is very hard for an attacker to obtain a profile of the user.

Authentication: Authentication protects from unauthorized service access and ensures that the user pays only his/her phone costs. However, the GSM network provides only authentication of the user to the network. The protocol does not identify the network to the user. Hence, false base station attacks are possible.

The second problem with authentication is based on the fact that the A3 and A8 algorithms are not specified in the GSM standard. Thus, the security of these algorithms relies on security-by-obscurity and therefore is not considered secure in general. The GSM standard committee recommended the use of COMP 128 for A3 and A8 which was broken in [4]. For this attack we need to perform 8 to 12 hours of calculations on the card to determine the user key K_i.

Key Establishment/Encryption Algorithm: Encryption protects user data; however, we need a stronger algorithm since the algorithm is broken and the key length is too short. In addition, encryption is only applied at the wireless interface (further security services are operator dependent). Thus, all information including TIMSI, RAND, SRES, K_s as well as the communication and signaling information are transmitted in clear within and between networks.

Transparency: Security features operate without user assistance. Thus, there is no indication to the user that encryption is activated. No explicit con-

firmation of properly accomplished authentication and correct authentication parameters is given when subscribers roam etc.

Channel Hijack: Protection against radio channel hijack relies only on the encryption mechanisms. However, encryption is not strong enough or is not used at all in some networks.

Inflexibility: The security functionality is inadequate and not flexible to upgrade over time. There will always be security holes in practical applications and, therefore, it is important to have mechanisms to upgrade cryptographic primitives or flawed software.

3 Universal Mobile Telecommunication System (UMTS)

The Universal Mobile Telecommunication System is envisioned as the successor to GSM. Hence, UMTS will be used in far-field communication, providing applications like DynRouting, Remote Diagnosis and Internet in the car. The main difference from GSM is that UMTS handles a higher data rate (up to 2 Mbps) per mobile user and that most security limitations of GSM are no longer present.

3.1 Overview

The UMTS standard is an extension of existing networks introducing the new components UTRAN, RNC, and Node B. In the following we describe the functionality of the new components in more detail. All the other existing components such as the HLR can be extended for UMTS. The handsets must be developed from GSM. The UMTS User Equipment (UE) is separated from the Mobile Equipment (ME) and the UMTS Subscriber Identity Module card (USIM), as in the GSM network.

UTRN and RNC: UTRAN is subdivided into individual Radio Network Systems (RNSs). The RNC provides control for each RNS. One or more Node B elements are connected to the RNC. RNC provides central control for the RNS elements, and handles protocol exchanges, central operation, and maintenance.

Node B: The Node B provides the radio transmission and reception. The main task of the Node B is to connect the UE (via radio interface) and the RNC (via asynchronous transfer mode). In addition, the Node B takes part in the power control of the UE.

3.2 Security Analysis

The security of UMTS is built on the security of GSM. The designers adopted the security features from GSM that have been proven to be needed and robust. On the other hand UMTS tries to ensure compatibility with GSM

in order to ease inter-working and hand-over. Furthermore, UMTS corrects problems with GSM and adds new security features necessary to secure new services, e.g. authentication of the network. In the following we give a brief description of the main security features of UMTS.

UMTS Authentication and Key Agreement (UMTS AKA): Authentication and key agreement in UMTS is similar to the one used in GSM [2]. The challenge-and-response protocol used in UMTS is enhanced to be able to provide mutual authentication. Figure 3 illustrates the performed protocol. After the VLR (SGSN) requested the authentication vector (AV), the AuC responses with the generated quintets ($RAND$, $XRES$, CK, IK, $AUTH$). The AUTH contains a sequence number (SEQ), a message authentication code (MAC) and an authentication management field (AMF).

In a second step the network can challenge the user, by sending a random number ($RAND$) and the $AUTH$ to the USIM. The user has the possibility to verify the received challenge data. This data could only have been constructed by someone possessing the secret key K_i. The USIM can also verify the freshness of the data by checking the SEQ. If all the tests are successful the network has authenticated itself to the user. The USIM can then generate the confidentiality key (CK), the integrity key (IK), and the response (RES). At the end the user sends the RES to the network and if $XRES$ equals RES, the two parties can start to communicate securely.

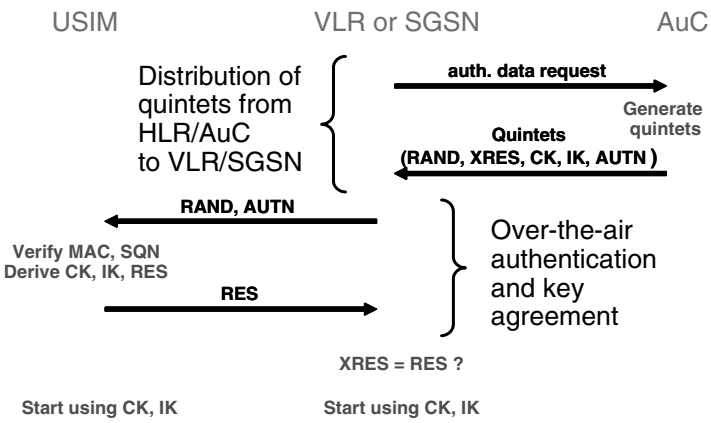

Fig. 3. UMTS message flow [18]

Confidentiality and integrity protection: In UMTS the security terminates in the RNC, which is the reason for the RNC being located closer to the core network than the Node-B (base-station). Using the established keys (IK and CK) one can start the confidentiality and integrity protection services. The confidentiality protection applies to both user data and the associated system signalling data using the $f8$ algorithm, realized by KASUMI algorithm [3].

The algorithm to provide integrity is called $f9$ and is also realized with the KASUMI algorithm. Note that the UMTS restricts integrity protection to the system signalling. Thus, the user data is not integrity protected. Another limitation of the integrity protocol in UMTS is that only a 32-bit long integrity check value is used.

Security in the UMTS core network: Network Domain Security (NDS) is needed to secure all important control plane protocols in the core network. There exists already a protection suite for the IP-based protocols, namely IPsec. Hence, this was the natural choice for security protection and furthermore it is implemented in the network layer and, thus, no changes are required to the target protocols [1].

4 Wireless LAN

Wireless Local Area Networks (WLANs) are based on a standard defined by the Institute of Electrical and Electronics Engineering in 1997 (IEEE 802.11) [11]. WLANs offer the possibility to easily set up networks and extend cable-based networks. Easy maintenance, flexibility and small devices make WLANs in particular interesting for embedded applications. Furthermore, WLANs have emerged in temporary networks (exhibitions, ad-hoc networks) or airports and downtown areas (hot spots).

Most current WLAN products are based on the standard IEEE 802.11b from 1999. The so called Wi-Fi Alliance certifies interoperability of Wireless Local Area Network products based on the IEEE 802.11 specification. Currently the Wi-Fi Alliance has over 200 member companies from around the world, and over 1250 products have received Wi-Fi [19].

Since 2001, major security problems of the standard are known and, finally, in 2004 a new standard addressing all security limitations was published (IEEE 802.11i). Unfortunately, most circulating products are not able to upgrade to the new standard.

The following two sections provide a brief overview of the standard with emphasis on security aspects. For detailed technical information and on WLAN security, the interested reader is referred to [6] and [11].

4.1 Overview

Radio Technology:

Wireless LAN systems approved in Europe use the ISM (Industrial-/Scientific-/Medical-) band between 2.4 and 2.48 GHz. It can be used free of charge and without any extra approval. The maximum transmission power is restricted to 100 mW. IEEE 802.11 specifies data transmission via Spread Spectrum with Frequency Hopping (FHSS) or Direct Sequence (DSSS). In the German 2.4 GHz band, 13 frequency channels with a bandwidth of 5 MHz are available. Three channels can be used simultaneously without interference (see Fig. 4).

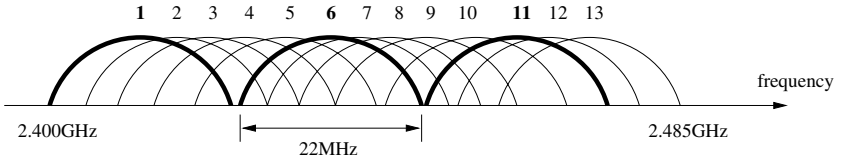

Fig. 4. WLAN channels

A single radio cell has a range between 10 and 150 m, depending on the environment, e.g., allowing for in-car and car-to-car communication at high data rates. Future radio systems (802.11n) will use the 5 GHz band with 19 approved channels (no overlap) in total.

Maximum throughput is 11 Mbit/s (802.11b) and 54 Mbit/s (IEEE 802.11g). Note that IEEE 802.11 does not guarantee a certain bandwidth since the maximum throughput depends on the number of clients and the quality of the radio link.

Network Architecture

IEEE 802.11 specifies two different architectures for wireless LANs: ad-hoc mode (IBSS=Independent Basic Service Set) and infrastructure mode (BSS=Basic Service Set). In ad-hoc mode, two or more (mobile) clients communicate directly over a radio link, whereas in the latter mode all communication is centralized via an Access Point (AP).

The infrastructure mode allows for roaming: An overlapping radio network with many access points can be constructed. The connection to a client can be retained while the client moves from one radio cell to another.

IEEE 802.11 entitles the union of many BSS as an Extended Service Set (ESS), with the connection network as Distribution System (DS). When a client enters the range of one or more APs, the APs broadcast a signal including a Service Set Identifier (SSID). The best AP in terms of signal strength is selected and the client turns to the AP channel. A client periodically surveys all channels in order to check for stronger or more reliable APs.

4.2 Security Analysis

Old standard IEEE 802.11

Security mechanisms are defined in IEEE 802.11 and, recently, in IEEE 802.11i. IEEE 802.11a, b, g and h do not describe additional security mechanisms. In the following, all mechanisms of the original standard IEEE 802.11 are outlined. Note that all mechanisms are fundamentally flawed and do not provide reliable security for sensitive information.

- (E)SSID: The (Extended) Service Set Identity provides network names and is always sent in clear and thus, can be eavesdropped. Users can allow for any (E)SSIDs or only for certain (E)SSIDs. Access points usually broadcast (E)SSID unless configured otherwise, which is not always possible. Even if disabled, SSIDs can still be obtained from (secondary) control information sent by APs.
- Media Access Control (MAC) Address: Though not specified in the standard, APs can grant access for only certain MAC addresses. In this case all access lists have to be edited by hand. Furthermore, MACs can easily be manipulated in wireless environments such that solely MAC filtering is no security enhancement.
- Wireless Equivalent Protocol (WEP): The goal of WEP was to provide equivalent security as in cable networks, including
  - Integrity: For each data packet, a 32-bit checksum (CRC32) is computed and attached to the data (ICV=Integrity Check Vector), as shown in Fig. 5. With encryption enabled, the data packet with ICV is encrypted. After decryption, the receiver checks the ICV. CRC32 can detect (transmission) errors with high probability. Cleverly swapped data bits cannot be detected since due to the linearity of CRC and the simple XOR encryption, the CRC can also be manipulated.

Fig. 5. WEP integrity check

  - Authentication: Two authentication modes are possible in connection with WEP encryption: "Open" with no authentication and "Shared Key". The latter mode accomplishes a challenge response protocol. The AP generates 128 pseudo random bits and sends them to a client (challenge). The client encrypts the challenge and sends the encrypted bits back to the AP (response). The client has authenticated properly if the

AP can decrypt the response. Note that the authentication is unilateral, i.e., the AP does not authenticate itself to the client.
This authentication protocol is completely flawed: An attacker can record an authentication step, and build the XOR of challenge and response. The resulting bits are the first bits of the key stream; thus, the attacker can authenticate himself from now on. Furthermore, messages can be faked with the help of computed key stream bits.

– Confidentiality: The key together with an initialization vector (IV) generates a pseudo random bit stream (stream cipher RC4) and XORs the bit stream with the data bit stream. Encryption and decryption are similar operations. The IV changes with every packet to ensure non-deterministic encryption. After encryption, the IV is appended in clear to the cipher text (see Fig. 6).

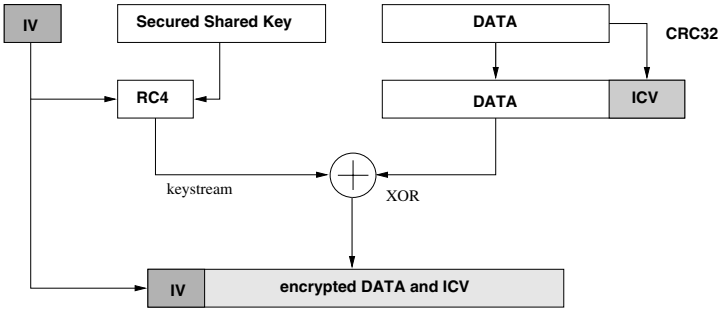

Fig. 6. WEP encryption

The IV is far too short since 24 bit yield only approximately 16.8 million different IVs; thus, if created at random, repetitions will occur every 4000's IV on average.
Two key lengths, 40 bit and 104 bit, are specified. Four 40-bit keys or one 104-bit key can be used. The same key is used for the whole network, and thus has to be provided to every client and AP. No key management is specified in IEEE 802.11. Keys are manually distributed and statically configured, implying infrequent changes. Thus, large volumes of traffic are encrypted with the same key, which lets cryptanalysts crack the key in a few days. Even the use of 40-bit keys with frequent changes is not advisable since conventional PCs can decrypt messages with "brute force" in a few days.

Additionally, RC4 suffers from some weaknesses which can be exploited by statistical tests [15]. The Internet offers many tools for automated attacks on WEP. Taking all aspects into account, WEP offers very little security.

New Security Standard IEEE 802.11i

The new security standard removes most security limitations of WEP. Before the approval of IEEE 802.11i in June 2004, the so-called Wireless Protected Access (WPA) was introduced as a stopgap method before the full standard. Many parts of the i-standard have already been introduced. The enhanced security standard allows for the following features:

- Mutual authentication: Client and AP authenticate each other.
- Temporary Key Integrity Protocol (TKIP): Each station gets a separate key for confidentiality, which is changed frequently.
- Extensible Authentication Protocol (EAP): Authentication involves a device proving its identity to another device. EAP enables authentication with an authentication server. The possibility of PKI-based authentication is given.

General Security Limitations of WLAN

- Uncontrolled propagation of radio waves: though specified, the range of WLAN can exceed 150 m depending on the quality of transmitter and receiver. Thus, eavesdropping is possible even beyond the range stated in the standard.
- WLAN (as well as any other radio-based system) can be jammed, regardless of any security service present.
- Profiling: Sending MAC addresses always in clear does allow for generation of profiles of mobile users.

5 Bluetooth

Bluetooth is an open standard (IEEE 802.15.1) for close-up range wireless voice and data communication [11]. The development of Bluetooth originated from the Bluetooth Special Interest Group in 1998 with currently more than 2500 manufacturers [5]. The specification is in version 1.1 at present; version 1.2 is close to approval.

5.1 Overview

Like WLAN, Bluetooth works with 79 channels in the ISM band (2400-2480 MHz). The data is modulated with Gaussian Frequency Shift Keying (GSFK) and transmitted in Time Division Duplex (TDD) in combination with Frequency Hopping Spread Spectrum (FHSS). A time slot is 625 s and the frequency changes 1600 times per second. The asynchronous connection less (ACL) transmission reaches a maximum throughput of 723.2 kbit/s in one direction and 57.6 kbit/s in the other (asymmetric) or 433.9 kbit/s in

both directions (symmetric). Voice transmission is accomplished with up to 3 synchronous connection oriented (SCO) channels 64 kbit/s each. Voice is encoded with Pulse Code Modulation (PCM) or Continuous Variable Slope Delta (CVSD) modulation.

The range of Bluetooth devices is 10-100 m depending on the transmission power (1-100 mW). To save power consumption, energy-efficient modes (sniff-, park-, and hold-mode) are specified. An adjustment of the transmission power is possible [6].

To guarantee the interoperability of all Bluetooth devices, different application profiles have been established: Generic Access Profile, Serial Port Profile, Generic Object Exchange Profile, Headset Profile, LAN Access Profile, Personal Area Networking (PAN) Profile, and more.

For identification, each device has a (worldwide) unique 48-bit hardware address (Bluetooth Device Address).

5.2 Security Analysis

Bluetooth specifies integrity protection and cryptographic mechanisms to ensure confidentiality and authentication. All mechanisms are implemented in hardware and, thus, are available at the data link layer. The cryptographic protocols are based on *link keys* negotiated during the *pairing*.

If two devices want to establish a secure communication, they have to be *paired*. Thus, a 128-bit *combination key* depending on both device addresses and random numbers is generated. For secure transmission of both random numbers, a 1- to 6-byte PIN has to be entered into both devices. If one PIN is fixed (as preset of the device) it has to be entered into the other device. Two devices with the same fixed PIN cannot be paired.

The standard allows for two other possibilities to generate session keys besides the use of the combination key:

- *Unit keys* are generated before the first use of a device (and cannot be changed). They are used if the memory of a device is constrained and to small for saving more keys or if a device has to be accessible to a huge group of devices.
- *Master keys* can be negotiated temporarily between devices if a master wants to use the same key with many devices. Master keys are only used in multi-point connections and are always sent to the clients encrypted with the session key.

Security Services

- Integrity protection: Integrity is protected with a CRC. As in the WLAN case, malicious manipulations cannot be detected if the CRC is also manipulated (see Section 4).

- Authentication: Authentication is based on a challenge response protocol using a symmetric cipher. Mutual authentication is achieved by two-sided unilateral authentication. The verifier sends a random number to the claimant, who generates a 32-bit response of the number together with the session key and his device number. From the same input data, the claimant generates another 96-bit *authenticated cipher offset*. This (secret) offset can be used as input for the generation of further encryption keys. The verifier simply generates the same 32-bit string and verifies the response.
- Confidentiality: Encryption is *always* optional and can be established if at least one party is authenticated. The encryption algorithm E0 is a stream cipher. For each data packet, a new initialization vector is used (non-deterministic). Encryption is only applied during radio transmission. Before broadcast and after reception the packets are not encrypted (no end-to-end encryption).

Further Security Limitations of Bluetooth

- Authentication is done by the devices, not by the user itself.
- Bluetooth Device Addresses can be manipulated (flash memory).
- Eavesdropping and recording of (unencrypted) voice transmission, is possible.
- Bluetooth can be jammed.
- Default settings are often insecure (PIN consists of zeros ...).
- The standard does *not* specify generation of random numbers.
- Man-in-the-middle attacks are possible even with authentication. Since encryption is implemented with a stream cipher, intercepted data can be manipulated if plain text (e.g., network address) is known.
- It is possible to intercept radio signals originating from Bluetooth devices (e.g. with a Bluetooth protocol analyzer ...).
- Unique device addresses simplify generation of profiles. Device addresses are not only used during the setup phase, but also attached to most data packets.
- Achieved security does not exceed 84 bit, though a 128-bit key can be used [16].

For a detailed description of major Bluetooth weaknesses, refer to [12].

6 Further Wireless Standards

Besides the widespread standards GSM, UMTS, WLAN, and Bluetooth, there exists a variety of other interesting wireless solutions, with potential application to in-car and inter-car communication. The following sections provide a short overview of ZigBee, DECT, HiperLan/2, and IrDA.

6.1 ZigBee

The application focus of ZigBee is monitoring and control. Its goal is the provision of wireless communication for very small devices (e.g., sensors). Parts of ZigBee are standardized up to the network layer in IEEE 802.15.4. The application layer is defined in the ZigBee Alliance [21]. Therefore, ZigBee should not be used as a synonymous standard for IEEE 802.15.4.

ZigBee uses 16 channels in the 2.4 GHz band with a throughput of up to 250 kbit/s via Offset Quadrature Phased Shift Keying (O-QPSK). Additionally, one channel with 20 kbit/s is available at 868 MHz via Binary Phase Shift Keying (BPSK). Transmission is accomplished using the Direct Sequence Spread Spectrum (DSSS) together with the CSMA/CA protocol (see Section 4). ZigBee allows for similar modes of networking to Bluetooth. Additionally, ZigBee specifies self reorganizing networks. Major technical properties of ZigBee are low power consumption ($\approx$0.5 mW) and medium range from 10-100 m. Good data integrity is achieved through high redundancy and dynamic channel selection.

The standard specifies a security toolbox for optional security functions such as authentication and encryption. The security services provides 32- to 128-bit AES. Key management is not specified.

6.2 DECT

Digital Enhanced Cordless Telecommunications (DECT) obeys the official ETSI standard for mobile communication networks for voice and data [8]. DECT is implemented in cordless telephones and can usually be found in offices, premises and private homes. DECT can also be used for bridging small distances (1-2 km) between provider and client. For interoperability between different DECT devices, a General Access Profile (GAP) is specified. Interworking Profiles define interfaces to other networks such as ISDN. Mobile networks can be built from DECT devices. Different communication services can be found in so-called Application Profiles for specific applications. The DECT Packet Radio Service (DPRS) and DECT Multimedia Access Profile (DMAP) allow for connections with high throughput comparable to Bluetooth.

In Europe, DECT operates at 10 carriers in a reserved band from 1880-1900 MHz with Frequency Division Multiplex (FDM). Each carrier is time division multiplexed (TDMA) in 24 slots. Time Division Duplex (TDD) with 12 duplex channels per carrier yields 120 available duplex channels in total.

DECT supports different modes of operation:

- Single cell system: The whole DECT system consists of a fixed part and a portable part (e.g., cordless telephone and base station).
- Direct mode: Two DECT portable parts communicate directly with each other.
- Multi-cell system: DECT is multi-cell capable and supports roaming.

The standard provides security mechanisms against eavesdropping. Authentication is based on a challenge response procedure with a 128-bit long-term key which has to be entered into both devices at the beginning. A portable device has to authenticate itself against the fixed part. Mutual authentication (i.e., additional authentication of a fixed part against a portable part) is optional. Detailed information about the authentication specification can be found in [8]. The authentication algorithms A11 and A12 are not publicly available and the strength of the algorithms is not known.

The implementation of the key management has several degrees of freedom, i.e., it is possible to register a cordless phone at a base station without previous exchange of a long-term key.

Encryption of the transmitted data is optional and is realized with a stream cipher. The stream cipher is initialized with a 64-bit cipher key (CK) and an initialization vector (IV). In practice, encryption is almost always toggled off or not even implemented. If encryption is implemented, usually key sizes of 64 bit are used, which is considered as insufficient nowadays. As with the authentication algorithms, the encryption algorithm is not publicly known.

Unattached encryption, protocol analyzers allow for profiles of base stations (e.g. how often encrypted session are established etc.). Built-in baby phone functionality enables easy eavesdropping.

6.3 HiperLAN/2

High Performance Radio Local Area Network Type 2 (HiperLAN/2) is a standard of the European Telecommunications Standards Institute (ETSI) [9]. HiperLAN/2 is a competitor of IEEE 802.11. IEEE 802.11 can be seen as wireless Ethernet whereas HiperLAN/2 works like a wireless ATM. Media access is centralized, connection oriented and supports quality of service.

HiperLAN/2 uses the 5 GHz band from 5.15 to 5.35 GHz and 5.47 to 5.725 GHz with 19 different channels of 20 MHz each. Maximum throughput is 54 Mbit/s at a range of approximately 30 m indoor and 150 m outdoor. The transmission power is limited to 200 mW indoors and 1 W (EIRP) outdoors. Handover of mobile stations to different base stations is supported by HiperLAN/2 to supply large areas. Transmission uses Orthogonal Frequency Division Multiplex (OFDM) modulation and is time multiplexed (TDMA/TDD, Time Division Multiple Access, Time Division Duplex). Carrier access handling is centralized by the base station or a dedicated mobile station which assigns time slots of different lengths.

Establishing a communication with a mobile station requires the MAC address of the base station. Different cryptographic options are possible: encryption with a common derived key, mutual authentication, and encryption with multi-cast keys provided by the base station. For each application, different keys are used. The session key for encryption is derived via Diffie-Hellman key agreement [7] and yields a DES or 3DES key. Weak and semi-weak keys are discarded. The base station initializes a new key exchange frequently. To

keep keys fresh, multi-cast keys are assigned frequently by the base stations, and are encrypted with the session key. Encryption implements DES or 3DES in Output Feedback (OFB) mode.

Authentication keys are either predistributed symmetric keys ($\geq$128 bit) or asymmetric key pairs (RSA512, RSA768 or RSA1024). Key management with certificates and a Public Key Infrastructure (PKI) is possible. Authentication is realized with a challenge response protocol. The response is generated either with a MD5-HMAC or with an RSA signature according to PKCS#1 v1.5 [14]. Security risks arise with the possibility of a man-in-the-middle attack on stations, which have no direct connection (see [6]). Anyway, HiperLAN/2 is much more secure than IEEE 802.11 (see Section 4). Dynamic MAC addresses assigned by the base station avoids profiling.

6.4 IrDA

The Infrared Data Association (IrDA) is a non-profit organization and released the first specifications of a protocol for an infrared interface in 1994 [10]. This infrared interface was designed as a wireless alternative to the serial port. IrDA uses wavelengths in the range of 850-900 nm and defines data rates up to 16 Mbit/s. The average range is 0.2-2 m and depends on the transmission power. Data integrity is achieved with CRC16 (up to 1.152 Mbit/s) and CRC32 (above 1.152 Mbit/s). For a successful communication, both devices have to face each other. IrDA does not specify any cryptographic service; thus, no authentication and encryption is possible.

7 Conclusion

Wireless networks can have many advantages compared to cable-based solutions: they are easy to set up and easy to maintain, they can be implemented in devices of nearly any size and can have little power consumption. The variety of the presented solutions demonstrate a wide possible field of application. High-end wireless ATMs (HiperLAN/2) are just as feasible as ultra-low-power transmissions in the kbit range (ZigBee). Thus, nearly any cable-based network has its wireless counterpart. Table 1 gives a short summary of the technical properties and the security of all presented systems (see also [20]).

Nevertheless, wireless protocols still have and will always have some serious penalties. Every wireless connection can be jammed, regardless of applied security services such as encryption. Thus, wireless devices should never be used for sensitive applications, e.g., applications which might harm or injure persons in cars in case of failure. Furthermore, the broad range of several wireless systems do not allow for clear localized restrictions of networks. Well-equipped attackers might always be able to eavesdrop any communication sent over the network. Strong cryptographic primitives and a diligent implementation of such algorithms and protocols can help to solve this disadvantage. Many

Table 1. Technical comparison of selected wireless standards and security features [20]

Market name Standard	GPRS/GSM 1xRTT/CDMA	UMTS 3GPP	Wi-Fi IEEE 802.11 a,b,g,h,i	Bluetooth IEEE 802.15.1	ZigBee IEEE 802.15.4
Application focus	Far field comm.	Far field comm.	Car-to-car comm.	In-car comm.	In-car comm.
System resources	16 MB+	16 MB+	1 MB+	250 kB+	4-32 kB
Battery life (days)	1-7	1-7	0.5-5	1-7	100-1000+
Network size	1	1	32	7-250	255/65,000
Bandwidth (kB/s)	56-128	56-14,000	800-54,000	720	20-250
Transmission range (m)	1,000+	1,000+	10-100	1-10+	1-100+
Success metrics	Reach, quality	Reach, quality, speed	Speed, flexibility	Cost, convenience	Reliability, power, cost
Security services: - Integrity - Authentication - Confidentiality	 X unilateral (X)	 X mutual X	 X mutual (opt.) X (opt.)	 X mutual (opt.) X (opt.)	 X mutual (opt.) X (opt.)

common wireless systems still suffer from poorly designed security services (e.g., old IEEE 802.11, GSM). Using these systems demands security solutions beyond the (flawed) standards, e.g., virtual private networks (VPNs) or other end-to-end encryption mechanisms.

References

1. 3GPP. Network Domain Security; IP network layer security. 3G TS 33.210, 2004. `www.3gpp.org`.
2. 3GPP. Security architecture. 3G TS 33.102, 2004. `www.3gpp.org`.
3. 3GPP. Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: $f8$ and $f9$ Specification. 3G TS 35.201, 2004. `www.3gpp.org`.
4. M. Briceno, I. Goldberg, and D. Wagner. GSM Cloning. 2003. `www.isaac.cs.berkeley.edu/isaac/gsm-faq.html`.
5. Bluetooth Membership Site. `www.bluetooth.org`, 2004.
6. BSI - Bundesamt für Sicherheit in der Informationstechnik. Drahtlose lokale Kommunikationssysteme und ihre Sicherheitsaspekte. Technical report, Projektgruppe "Local Wireless Communication", 2003.
7. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
8. European Telecommunication Standards Institute. ETSI EN 300 175-1, Digital Enhanced Cordless Telecommunications (DECT), Common Interface (CI) Part 1-8, 1992.
9. European Telecommunication Standards Institute. ETSI TR 101 683, HIPERLAN Type 2 Overview, 1992.
10. Infrared Data Association Site. `www.irda.org`, 2004.

11. Institute of Electrical and Electronics Engineering. IEEE 802 LAN/ MAN Standards. `standards.ieee.org/getieee802`, 2004.
12. M. Jakobsson and S. Wetzel. Security weaknesses in bluetooth. In *Proceedings of the Cryptographer's Track at the RSA Conference (CT-RSA 2001)*, LNCS 2020. Springer, September 2001.
13. Wolfgang Rankl and Wolfgang Effing. *Handbuch der Chipkarten*. Hanser-Verlag, 1966.
14. RSA Laboratories, 1993.
15. A. Shamir, S. Fluhrer, I. Mantin. Weaknesses in the Key Scheduling Algorithm of RC4. In *Selected Areas in Cryptography - SAC 2001*, volume 2259 of *Lecture Notes in Computer Sciences*, pages 1-24. Springer-Verlag, 2001.
16. S. Lucks and S. Fluhrer. Analysis of the E_0 Encryption Scheme. In *Selected Areas in Cryptography - SAC 2001*, volume 2259 of *Lecture Notes in Computer Sciences*, pages 38-48. Springer-Verlag, 2001.
17. Klaus Vedder. Gsm: Security, services, and the sim. In Bart Preneel and Vincent Rijmen, editors, *State of the Art in Applied Cryptography*, volume 1528 of *LNCS*, pages 224–240. Springer-Verlag, 1997.
18. M. Walker. On the security of 3gpp networks. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 102–103. Springer, 2000.
19. WiFi Aliance. `www.wi-fi.com`, 2004.
20. J.F. Wollert. Bluetooth, WLAN und ZigBee für die Automatisierungstechnik. *etz – Elektrotechnik und Automation*, 6:10–18, 2004. VDE Verlag.
21. ZigBee Alliance Site. `www.zigbee.org`, 2004.

Embedded Cryptography: Side Channel Attacks

Kai Schramm, Kerstin Lemke, and Christof Paar

Horst Görtz Institute for IT Security
Ruhr-Universität Bochum, Germany
Universitätsstrasse 150
44780 Bochum, Germany
{schramm, lemke, cpaar}@crypto.rub.de

Summary. This article gives an overview of state-of-the-art side channel attacks and corresponding countermeasures which are currently discussed in the scientific literature. We compare different attacks with respect to the algorithms attacked, measurement costs, efficiency and practicability. In particular, we examine simple power analysis (SPA), differential power analysis (DPA), internal collision attacks and template attacks. Moreover, we give a brief overview of various countermeasures based on hard- or software which should always be implemented in order to secure cryptographic algorithms against side channel attacks.

Keywords: side channel attacks, SPA, DPA, internal collisions, template attacks

1 Introduction

Cryptographers have traditionally designed new cipher systems under the assumption that the system would be implemented in a closed, reliable computing environment which does not leak any sensitive information. Hence, implementations were generally regarded as black boxes which have an input and an output interface only. However, in 1996 Paul Kocher et al. showed that various public key ciphers were vulnerable to attacks which analyze the execution time of an algorithm [20]. We will investigate timing attacks in more detail in Section 2. Two years later in 1998 they were able to show that the power consumption of smart cards available at that time leaked sensitive details about the executed operations and data being processed [18]. Attacks which analyze these unintentional sources of information are generally not considered in black box designs of cryptographic algorithms.

In general, cryptographic algorithms are either implemented as software running on a processor or as hardware, e.g. an application-specific integrated circuit (ASIC) consisting of semiconductor logic gates and memory elements.

Various classes of attacks which try to compromise the implementation of a cryptographic algorithm are shown in Figure 1. This article is focused on *side channel* attacks, which investigate and analyze the timing behavior [20, 17], the power consumption [18, 19, 2, 22, 24] or electromagnetic emission [1] of an implementation.

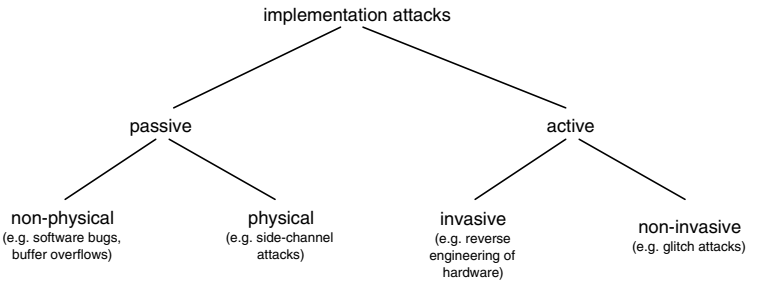

Fig. 1. Various classes of implementation attacks

Due to the physical nature of hardware and software implementations, there will usually be a leakage of side channel information, if no adequate countermeasures are used. Hence, even though side channels are not regarded in the black box design of an algorithm they do play an extremely important role in practical implementations. This is shown in Figure 2. It is widely known that most smart card manufacturers are aiming to eliminate side channel information as much as possible, e.g., by means of hardware and software countermeasures [6].

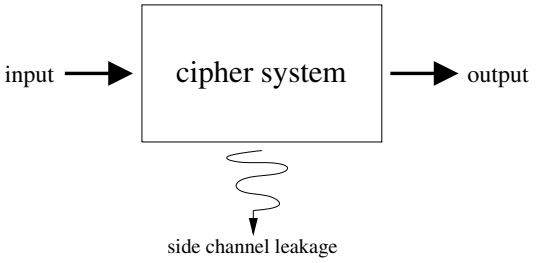

Fig. 2. Information leakage through a side channel of a cryptographic system

Paul Kocher et al. demonstrated in the original paper [18] that common algorithms such as the Data Encryption Standard (DES) and Advanced Encryption Standard (AES) implemented in smart cards could be broken by analyzing the power consumption. A cipher executed by a microprocessor usually

influences the behavior of many thousand *Complementary Metal Oxide Semiconductor* (CMOS) logic gates. In Figure 3 a simple CMOS inverter consisting of a *p-channel Metal Oxide Semiconductor* (PMOS) and an *n-channel Metal Oxide Semiconductor* (NMOS) transistor is shown.

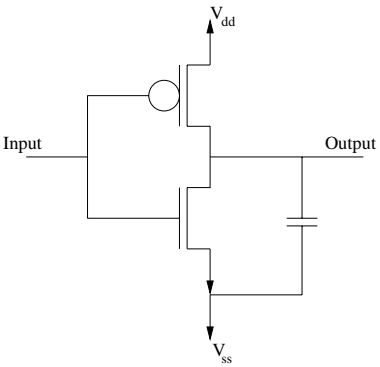

Fig. 3. CMOS inverter

In static operation the voltage levels of the input and output node will be either at V_{dd} (high) or V_{ss} (low), however, during a transition of the input voltage from V_{dd} to V_{ss} or vice versa the inverter will act as a short circuit for a short time resulting in an increased current flow.

The power consumption of a microchip is typically analyzed by putting a small shunt resistance R_s (e.g., a few 10 Ω) between the V_{dd} (V_{ss}) pin of the microchip and the true source (ground). Another approach which decouples the oscilloscope from the target hardware is to use a current probe which measures current induced by the power line of the target hardware. In the former case noise induced by a common power source can be reduced by replacing the voltage source with a well-filtered source with low-voltage ripple. A digital oscilloscope with low quantization noise (e.g. a 12-bit Analog Digital Converter) and high sampling rate (e.g., 500 MHz) is then used to digitize the voltage over the shunt resistance, which is proportional to the current drawn by the microchip. This is shown in Figure 4.

In [18], two practical attacks, Simple Power Analysis (SPA) and Differential Power Analysis (DPA), were introduced. In SPA attacks an adversary has previously obtained knowledge of the target hardware and knows details such as timing offsets of the particular implementation. SPA makes use of characteristics that are directly visible in one measurement trace. (If the noise is a dominant source, alternatively, an average trace using the same input data can be used.) The secret key needs to have some simple, exploitable relationship with the operations that are visible in the measurement trace. Therefore, very often vulnerable implementations use key-dependent branching in the

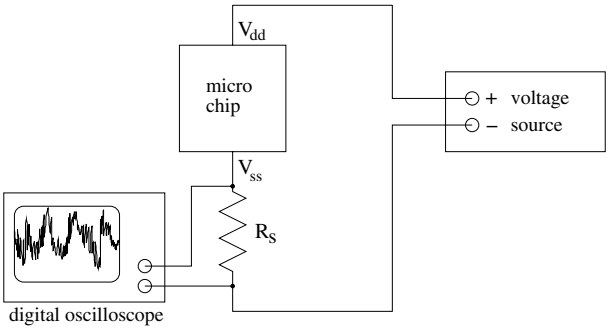

Fig. 4. Power analysis of a micro chip

source code which can be detected with side channel analysis. The adversary measures the particular side channel and thus deduces information about specific instructions, their opcodes, operands and addresses affecting the internal busses of the target hardware. SPA is discussed in more detail in Section 3.

The main idea of DPA is to detect regions in the side channel information of a device which are correlated with the secret key. However, unlike simple power analysis DPA further processes the measured power (or electromagnetic) traces by using statistical methods. First, an adversary starts encrypting a certain number of various uniformly distributed plaintexts and measures the corresponding power traces. Next, the adversary sets up key hypotheses about the secret key and predicts the value of a chosen intermediate key-dependent variable for each measurement trace. Finally, every trace is correlated with the intermediate variable[1] and the correlation signal is analyzed. If the correlation signal contains distinct peaks for a particular key hypothesis, this key hypothesis was correct. The most important advantages of DPA are twofold: the adversary does not need to know specific details about the examined hardware or about the particular implementation, as e.g. precise timing information. DPA is discussed in further detail in Section 4.

Recently, attacks which analyze side channel information in order to detect internal, key-dependent collisions have been published [33, 31, 34]. Unlike DPA internal collision attacks are chosen plaintext attacks and partially resemble methods used in differential cryptanalysis. The basic idea is to cause an internal collision for some key-dependent intermediate variable at some point within the algorithm and observe such a collision with side channel analysis. Depending on the algorithm, collision attacks may be advantageous in terms of measurement costs. For example, in the case of the AES collision attack published in [31] only 40 chosen plaintext encryptions are required to determine the entire 128-bit key. Collision attacks are discussed in the first part of Section 5.

[1] or a function of the intermediate variable, e.g. its Hamming weight

In the second part of Section 5 template attacks which were originally proposed by Rao et al. in [8] are discussed. Templates attacks are based on multivariate analysis of side channel measurements, i.e. in contrast to classical SPA or DPA attacks a set of multiple signal points of a side channel trace (and the statistical behavior among these points) is analyzed. These attacks require an adversary to generate key-dependent templates using a device identical to the target hardware. The main advantage of template attacks over SPA/DPA and collision attacks is the fact that once templates have been obtained from the test device only a single measurement of the identical target device is required to determine the secret key used in the target device. Thus template attacks are ideally suited for ciphers which use ephemeral keys such as stream ciphers, since stream ciphers are often used in protocols which update the key frequently. Finally, in Section 6 an overview of current software and hardware countermeasures which thwart side channel attacks is given.

2 Timing Analysis

In 1996 [20], timing analysis was the first side channel based attack in the public literature. Paul Kocher described this methodology to compromise keys of RSA, DSS and other cryptosystems by measuring the execution time of the overall cryptographic operation.

The basic algorithm that is needed for these cryptographic algorithms is modular exponentiation. To perform a modular exponentiation $c = a^b \bmod m$ in $\mathbb{Z}_m$, the bitwise representation $b = [b_{n-1}b_{n-2}\cdots b_1 b_0]$ is used. The *square and multiply* algorithm evaluates this representation, e.g. by starting from the most significant bit b_{n-1} .

```
c := 1
for k := n-1 down to 0 do {
  c := c*c mod m
  if b[k]=1 then c := c*a mod m
}
return c
```

Fig. 5. Square and multiply algorithm.

Obviously, the time needed to process one exponent bit is increased if the bit is set to '1' because of the additional multiplication. Note that the original publication by Kocher does not exploit the observation of internal characteristics during the processing of the algorithm, e.g. the execution times of single multiplications. A straightforward implementation of the modular exponentiation as shown in Figure 5 is extremely vulnerable against Simple

Power Analysis (SPA, see Section 3) under the assumption that the squaring and multiplication operation can be distinguished (e.g. by their timing and power consumption pattern).

For a successful timing attack it is required that the execution time of the *square and multiply* algorithm is data-dependent. The timing attack can be applied both by an attacker who has physical access to the cryptographic device and by a remote attacker who can measure running times, e.g. by eavesdropping a network communication line. It is a statistical attack including multiple executions using varying input data a_i and a fixed secret key b. The overall execution time is measured which is denoted by $T(a_i)$ wherein $i \in \{0, ..., k-1\}$ runs through all k single measurements. It is assumed that the input data a_i as well as the modulus n are known. The adversary is trying to disclose the secret exponent b.

The attack requires that an attacker is able to simulate or predict the timing behavior of the attacked device rather accurately. If the input data to a multiplication or squaring operation is known, the execution time can be either simulated or directly measured. We call this second device needed the *simulating device*. For the explanation of the bit-by-bit approach we assume that the attacker has already compromised the first $n-j$ bits of b. For each input value a_i the attacker calculates the execution time for the first $n-j$ bits using the simulating device: $T_{n-j}(a_i)$. The statistical decision problem is now whether bit b_{n-j-1} equals 0 (hypothesis H_0) or whether b_{n-j-1} equals 1 (hypothesis H_1).

The execution time for the processing of the next bit is determined for both hypotheses H_0 and H_1 using the simulating device. For the hypothesis H_0 the simulating device returns the additional execution time $T_{H_0}(a_i)$. Accordingly, let $T_{H_1}(a_i)$ be the time for the hypothesis H_1.

The hypothesis test computes the variance (second empirical moment) of the data sets

$$H_0: \quad T(a_i) - T_{n-j}(a_i) - T_{H_0}(a_i)$$

and

$$H_1: \quad T(a_i) - T_{n-j}(a_i) - T_{H_1}(a_i).$$

As result, the test returns the hypothesis with the smaller variance for its data set on the basis of k timing measurements. The bits b_{n-j-2} to b_0 are not predicted and their contribution to the overall execution time is treated as an additional noise. The times for the modular operations are effectively independent from each other. Let $Var_M(t)$ be the variance for the multiplication and $Var_S(t)$ be the variance for the squaring operations. Then the overall variance is $n\ Var_M(t) + Ham(b)\ Var_S(t)$, wherein $Ham(b)$ gives the Hamming weight of the exponent b. By increasing the number of measurements k the effect of the noise contribution can be minimized. The minimum number of measurements k needed for the hypothesis test is proportional to the exponent bit size n.

Practical tests were done in [11] and [30]. Both publications use a Montgomery multiplication with a constant execution time except that, if the intermediary result of the multiplication is greater than the modulus, an additional subtraction has to be performed. In summary, there are only two possible execution times for each multiplication which simplifies the statistical model. For a 512-bit exponent [11] reported that approximately 350,000 measurements are needed, whereas the improved statistics of [30] reduced the number of samples by a factor of up to 50. Another variant of the timing attack can be applied at a CRT implementation of RSA, if implemented using Montgomery multiplication (see [27]).

Power analysis attacks (see Sections 3 and 4) are generally more effective than pure timing attacks as they reveal information about the internal processing. Combinations of them are feasible (see [28] and [32]).

3 Simple Power Analysis

Side channel attacks based on SPA are strongly implementation dependent. Some SPA scenarios exploit key-dependent branches of software implementations. These characteristics of an implementation can be obvious thus an attacker may succeed with a single power trace. A general guideline for defending against SPA is to avoid key-dependent branches.

In more advanced SPA scenarios the adversary must not only know details about the power consumption behavior of the target hardware, but also the exact points of time of the observed key-dependent instructions. In [22] it is stated that generally two types of information leakage have been observed in SPA: Hamming weight leakage and transition count leakage. In a precharged bus design [36] the number of zeros driven onto the bus is directly proportional to the amount of current that is being discharged from the driven gates. Hence, with circuits which use precharged busses it is possible to determine the Hamming weight on the data or address bus. Transition count information leaks when the dominant source of current is due to the switching of the gates that are driven by the bus: during a gate transition from high to low or low to high both transistors in Figure 3 will conduct current for a short time. This transition current consists of two parts [2]: a larger part, which arises from charging/discharging succeeding gates and parasitic capacitances, and a smaller part, which is due to the dynamic short circuit current between V_{dd} and V_{ss}. Theoretically, it is possible to distinguish between an output state change from high to low and low to high, because discharging succeeding gates will result in an increased transition current, while charging will result in a decreased transition current [22]. As shown in Figure 6, the relationship between power consumption and Hamming weight is approximately linear.

As explained above, SPA directly examines single power traces. Nevertheless, an attacker must have detailed information about the hardware and about the particular algorithm implementation. If attacking DES for example,

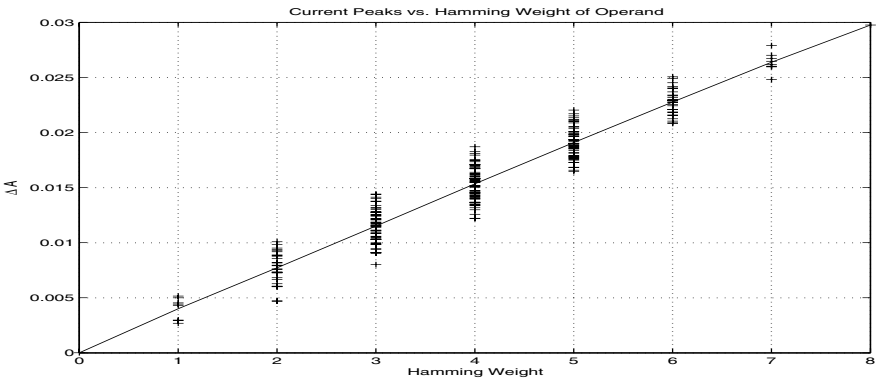

Fig. 6. Power analysis of a micro chip: the current consumption is approximately linear proportional to the Hamming weight of the processed operand

an attacker could analyze the power traces during the PC1 permutation in the key-scheduling algorithm.

Other occasions where implementations of cipher algorithms might be vulnerable to SPA are carry bit related instructions, such as shifting the key bytes or the use of conditional branches to test bit values [22]. String or memory comparison instructions typically perform a conditional branch when a mismatch occurs. This conditional branching can cause large SPA characteristics.

As a consequence, in order to thwart SPA attacks implementers should generally try to avoid key-dependent branches and single-bit instructions which process key-dependent bits. Moreover, the execution time of sub-functions should never depend on the key, but be constant in order to thwart timing attacks.

4 Differential Power Analysis

DPA is based on a simple but yet brilliant idea: an adversary sets up hypotheses using a reduced key space and computes the cross-correlation of each side channel measurement with a selection function which combines the known input data and the key hypothesis. The selection function can be an intermediate variable occurring within the analyzed algorithm depending on this hypothesis. Let us investigate a DPA attack against the popular block cipher DES to clarify this approach. First, a certain number N of power traces (or EM traces) must be collected thus an adversary generates (or observes) random plaintexts X_i, encrypts these plaintexts and measures the corresponding power traces $P_i[t]$. Hence, DPA is a known plaintext attack. If an adversary knows the plaintext and the key hypothesis, the adversary can calculate the input and by table look-up the output of the s-box functions in round one.

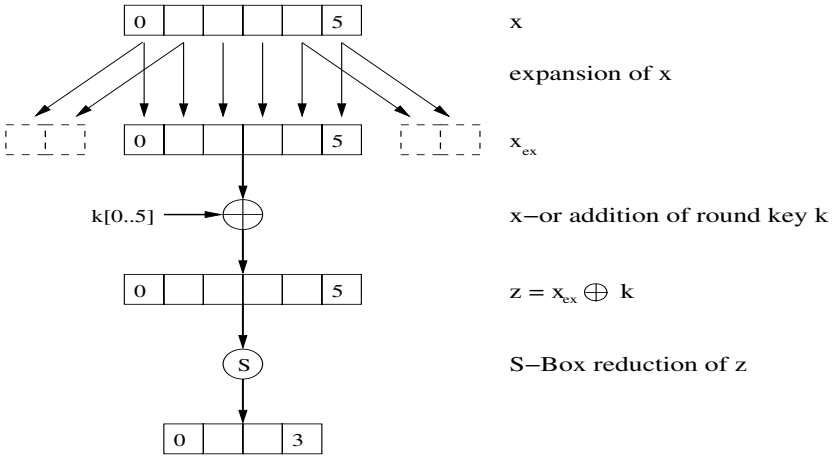

Fig. 7. Input and ouput of an s-box in DES

DPA of DES usually focuses on one s-box output at a time. Thus the overall key space of DES (56 key bits) is reduced to 8 times 6 key bits in the first round. The remaining 8 key bits can be revealed in the second round, if we deal with a Triple-DES implementation; otherwise they can be tried out by brute force. In DPA either a single bit of an s-box output or the entire s-box output (i.e. its Hamming weight) is analyzed. We will explain both methods in the forthcoming text. Since the key is secret, the adversary has to make a hypothesis K_s about 6 key bits of round key one in order to predict the input and output of a chosen s-box (see Figure 7). In single-bit DPA, the adversary is then able to predict the state of a chosen s-box output bit for every encryption. If the output bit is set, the corresponding side channel traces are assigned to a 1-partition, if it is cleared they are assigned to a 0-partition. The mean of both partitions is then computed. The mean curves are finally subtracted which results in a differential trace $\Delta_D[t]$, that contains significant peaks at those points in time when the predicted output bit has a leakage:

$$\Delta_D[t] = \frac{\sum_{i=1}^{N} D(X_i, K_s) P_i[t]}{\sum_{i=1}^{N} D(X_i, K_s)} - \frac{\sum_{i=1}^{N} (1 - D(X_i, K_s)) P_i[t]}{\sum_{i=1}^{N} (1 - D(X_i, K_s))} \quad (1)$$

The function $D(X_i, K_s)$ is called a selection function and in single-bit DPA gives the state (i.e. 1 or 0) of a chosen s-box output bit for a particular plaintext X_i and key hypothesis K_s. Because the adversary does not know the secret key, the differential trace has to be calculated for all $2^6 = 64$ possible key hypotheses. If the key hypothesis is correct, large spikes will occur at those points of time in the differential trace when instructions process the observed s-box output bit. However, if the key hypothesis is not correct, the differential trace will converge against zero for all points in time, since the predicted bit

state $D(X_i, K_s)$ is always uncorrelated with the corresponding power trace measurements $P_i[t]$. In Figure 8, exemplary two differential traces are shown. The first differential trace contains a large spike for a particular moment of time indicating a correct key hypothesis, while the other differential trace is dominated by noise for all times indicating a false key hypothesis.

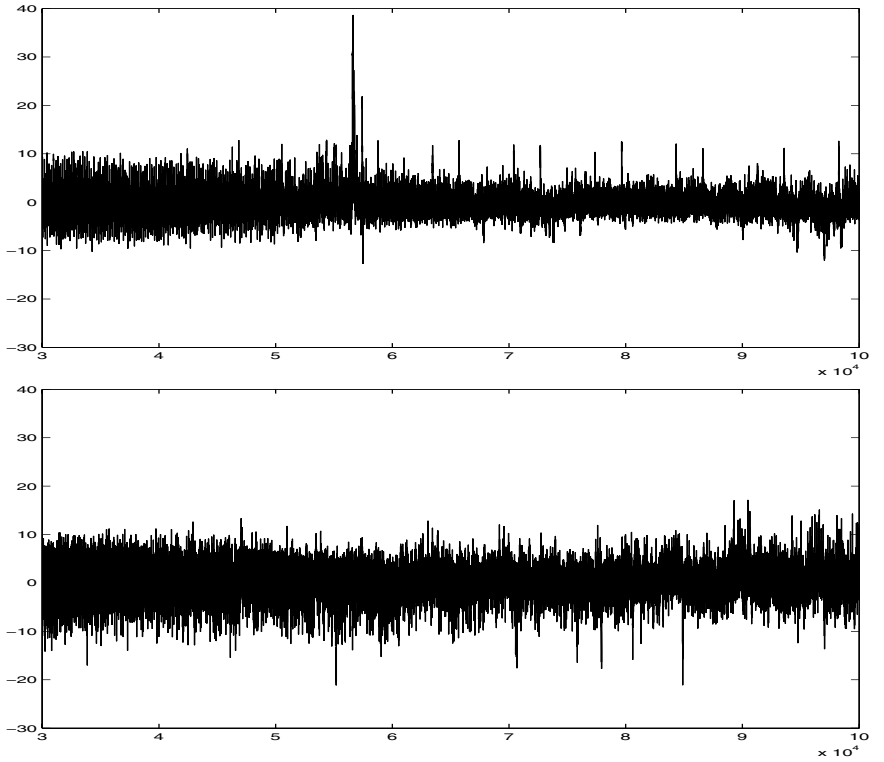

Fig. 8. DPA results: differential traces for the correct key hypothesis (first plot) and one false key hypothesis (second plot) as a function of time

As stated above, it is also possible to compute the cross-correlation factor of power traces $P_i[t]$ with the Hamming weight HW of the output of a chosen s-box $S_j(x_i)$. In this case the selection function is defined as

$$D(X_i, K_s) = HW(S_j(x_i)) - 2 \qquad \in \{-2, -1, 0, 1, 2\} \tag{2}$$

The resulting correlation signal is defined as [24, 21]:

$$c[t] = \frac{\sum_{i=1}^{N} D(X_i, K_s) P_i[t]}{\sqrt{\sum_{i=1}^{N} D(X_i, K_s)^2} \sqrt{\sum_{i=1}^{N} P_i[t]^2}} \tag{3}$$

As shown in [21], correlation of power traces with the Hamming weight can be advantageous if the output of a linear function instead of a non-linear function is analyzed in DPA. Typical examples of linear functions in block ciphers are the x-or addition of sub-keys with intermediate variables. However, whether single-bit or multiple-bit DPA is more successful depends on the particular target hardware architecture. An adversary will be more successful if the target device is analyzed prior to the attack.

5 Other Side Channel Attacks

5.1 Internal Collision Attacks

In cryptography the term *collision* denotes the case that some function results in an equal output for two different inputs. Cryptanalysts have generally used collisions to attack hash functions in the past [13, 4]. Most of the previous attacks against hash functions only attacked a few rounds, e.g., three rounds of RIPEMD [12, 25]. In [13], it was shown that MD4 is not collision free and that collisions in MD4 can be found in a few seconds on a PC. Another historic example of breaking an entire hash function is the COMP128 algorithm [3]. COMP128 is widely used to authenticate mobile stations to base stations in GSM (Global System for Mobile Communication) networks [16]. COMP128's core building block is a hash function based on a butterfly structure with five stages. In [4], it was shown that it is possible to cause a collision in the second stage of the hash function, which fully propagates to the output of the algorithm. Hence, a collision can be easily detected, revealing information about the secret key.

However, if an adversary is able to perform side channel analysis on a cryptographic target hardware another approach is possible, as well. The main idea is to detect internal collisions within a cipher by analysis of the power consumption or the electromagnetic radiation. Contrary to strictly cryptanalytic collision attacks, internal collisions are exploited, which are not necessarily detectable at the output. Various versions of internal collision attacks have been applied to several symmetric ciphers, such as DES and AES [29, 37, 33, 31, 34].

In [10], it was first shown that the f-function of DES is not one-to-one for a fixed round key, because collisions can be caused in three adjacent s-boxes. In [33] it was discovered that such internal collisions reveal information about the secret key. On average[2] 140 different encryptions were required to find the first collision; a significantly lower number of additional encryptions was required to find further collisions. The propagation path of a collision occurring in the f-function of round one of DES is shown in Figure 9. The authors examined the practicability of the collision attack with an 8051 software implementation of DES. Moreover, the authors claimed that no averaging of power

[2] Averaged over 10,000 random keys.

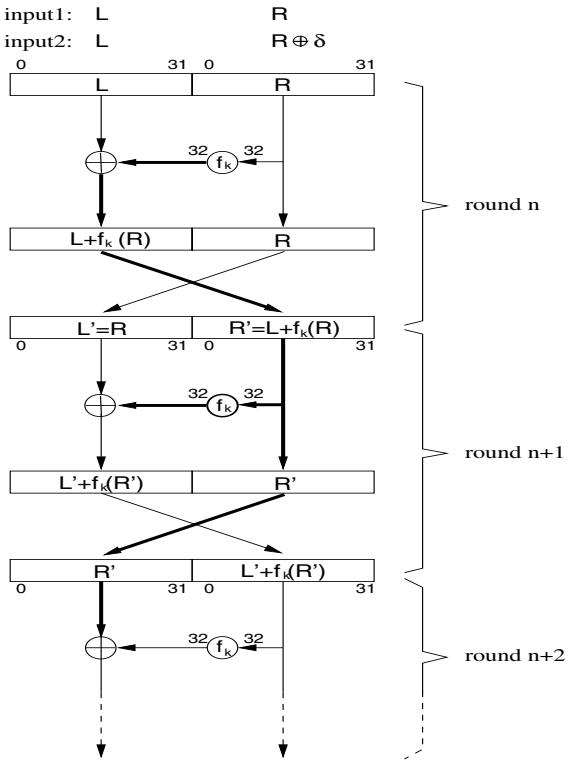

Fig. 9. Propagation path of a collision in DES

traces was necessary and that they used simple cross-correlation of traces to detect collisions.

Another more general attack against Feistel ciphers such as DES based on internal collisions, which requires fewer measurements, was originally discussed by Andreas Wiemers in [37] and later once again in [34]. This attack tries to exploit internal collisions at the x-or addition output within the Feistel cipher [37]. In the case of the DES collision attack, a single s-box is active at a time, i.e. an adversary changes the input and thus the 4-bit output of a single s-box in round one while the input of all remaining seven s-boxes is fixed. Next, the adversary varies the corresponding 4 bits of the left half input L of round one until a collision occurs at the output of the x-or sum. A collision in the x-or sum will again result in an equal input to round two of DES. If all eight s-boxes are attacked in a row, only $8 \times 8 = 64$ measurements are required this way.

In [31], an internal collision attack against AES was presented. The authors show that partial, key-dependent collisions also occur at the output of linear transformations, such as the mix column transformation of AES. In

order to detect internal collisions, the least-squares method was applied by the authors to compute the difference of two power traces [31]. By taking advantage of the birthday paradox, it is shown that it is possible to cause a collision in a single mix column output byte with as little as 20 measurements. The authors claim that whenever a SPA leak is present from which collisions can be determined with certainty, then each collision will reveal at least 8 bits of the secret key. Furthermore, in a parallelized approach, they show that it is possible to observe collisions in all four output bytes of the mix column transformation with an average of only 31 measurements, which results in knowledge of all 32 key bits. Finally, if this approach is applied to all four columns of the AES in parallel, it is possible to determine the entire 128-bit key with only 40 measurements. The authors claim that this is the biggest advantage of the AES collision attack over DPA. However, DPA has the advantage of being a known plaintext attack whereas the collision attack is a chosen plaintext attack. A DPA against an unprotected implementation of AES which yields the correct key hypothesis typically requires between 100 and 1000 measurements depending on the presence of superimposed noise. Under ideal circumstances a collision attack against AES might be possible with only 40 measurements, however.

5.2 Template Attacks

Stream ciphers are most often resistant against DPA attacks, since the internal state of stream ciphers generally evolves fast and an adversary is not always able to trigger a stream cipher with the same key in order to obtain enough samples to mount a successful attack. Unlike DPA, template attacks which consist of a profiling phase and a key extraction phase are based on the multivariate analysis of side channel traces and are ideally suited for stream ciphers. In [8], Chari et al. introduced template attacks and showed that it is possible to break the popular stream cipher RC4 with a single measurement. Template attacks implicate that an adversary has access to a programmable device identical to the target hardware during the profiling phase. The key-dependent model of captured side channel information for one operation is called a template and contains a noise-free signal for this particular operation and a noise characterization for that particular case. An adversary first analyzes the sample hardware and creates templates for parts of the key. The adversary then analyzes the target hardware and tries to classify the secret key using the stored templates. The goal is to significantly reduce the number of possible keys or even come up with the correct key. Building a template for all possible key values is of course not feasible. Therefore, a so-called extend and prune strategy has to be used. The idea is to use an incremental strategy. i.e. an adversary only builds templates for a small subset of the key and tries to break the cipher in a divide-and-conquer manner. At each step more and more bits of the unknown key are used for the next key hypothesis.

Every template consists of a signal model and a noise model for a particular key-dependent intermediate variable. During the profiling phase the templates are obtained from the test device. During the key extraction phase the precomputed templates are used to determine the state of the key-dependent intermediate variable with some error probability. Hence, first n power traces corresponding to a particular operation are measured from the test device. Modeling the noise-free signal is simply achieved by averaging these n power trace measurements P_j.

$$\overline{P} = \frac{1}{n} \sum_{j=1}^{n} P_j \tag{4}$$

In order to create a data-dependent noise model for each operation, the first step is to compute noise vectors N_j for every power trace measurement by subtracting the power trace from the averaged noise-free signal.

$$N_j = \overline{P} - P_j \tag{5}$$

In order to save processing resources the number of sample points t per averaged power trace $\overline{P}$ and noise vectors N_j should be cut down to a smaller number of significant points (e.g. between 20 and 50) , i.e. to those distinct peaks which are strongly dependent on the particular intermediate variable. Template attacks are based on the fact that the significant points of the noise vectors are drawn from different data-dependent probability distributions. As a result, recognizing the probability distributions of a noise vector by comparing an observed noise vector with previously computed noise vectors makes it possible to classify the corresponding signal. If it is assumed that all probability distributions are approximately normal distributions, a multivariate gaussian analysis of the noise vectors can be applied in order to determine the corresponding signal with maximum probability. In multivariate gaussian analysis a noise model is basically represented as a matrix of covariances of noise vectors N_j. Hence, each entry of the $t \times t$ covariance matrix CM is defined as:

$$\begin{aligned} CM(t_x, t_y) &= cov(N_j(t_x), N_j(t_y)) \\ &= \frac{1}{p-1} \sum_{j=1}^{p} (N_j(t_x) - \overline{N_j(t_x)})(N_j(t_y) - \overline{N_j(t_y)}) \end{aligned} \tag{6}$$

where the terms $\overline{N_j(t_x)}$ and $\overline{N_j(t_y)}$ denote the mean values of all noise vectors at particular points in time t_x and t_y.

Once a certain number of templates has been generated to classify the key-dependent intermediate variable an adversary can mount a template attack by measuring a single power trace P' of t sample points from the target device in order to find out which template describes the noise characteristic

of the measured power trace in the best way. First, he computes a noise vector $N_i = \overline{P}_i - P'$ for each template i. Next, the joint probability of noise vector N_i being contained in template i can be computed as

$$p(N_i) = \frac{1}{\sqrt{(2\pi)^t |CM_i|}} \cdot e^{-\frac{1}{2} N_i^T CM_i^{-1} N_i} \tag{7}$$

In [8] it is stated that this maximum likelihood approach of determining the correct template is optimal if data-dependent noise has a normal (i.e. gaussian) distribution. The major point of template attacks is the fact that only a single measurement is required in order to determine the correct corresponding template and thus the correct sub-key. The authors state that they were able to successfully break implementations of RC4, DES using power analysis and even an SSL accelerator card inside a closed server using EM analysis. Template attacks have only been discussed marginally in side channel related publications so far, probably due to their complexity. However, template attacks currently represent the state of the art attack to break stream ciphers or any cipher used in a protocol where only a single trace is available.

6 Countermeasures

As we have shown in the previous sections of this article side channel attacks represent a serious threat to cryptographic implementations. In this section we will discuss countermeasures which are generally implemented in hardware or software or a combination of both in order to thwart these attacks.

6.1 Protection against Timing Attacks

Since timing attacks exploit the fact that the duration of an algorithm is somehow correlated with the secret key, the easiest way to counteract these attacks is to make the algorithm time invariant, i.e. the duration of all computations must be strictly constant independent of the input and key. For example, an RSA implementation may be secured against timing attacks by rewriting the square-and-multiply algorithm, so that its duration is always constant independent of the Hamming weight of its exponent and the data processed [20]. However, this also implies an obvious performance drawback, since this duration would equal the time which is needed to en- or decrypt the all-one exponent 111...1. More efficient, although less generic, countermeasures can be implemented to thwart certain known timing attacks. For example, several countermeasures use additional subtractions in an RSA Montogomery multiplication [35]. In general, block ciphers are very easy to secure against timing attacks, since most subfunctions are usually time invariant. Furthermore, it has been shown that block ciphers which do contain time-variant subfunctions, such as RC5 and AES, can be easily secured [17] as well.

6.2 Protection against Power and EM Attacks

Software Countermeasures:

In general, there exist two classes of software countermeasures: the first class masks the processed data in order to make it uncorrelated with side channel measurements and the second class desynchronizes repeated side channel measurements by inserting random dummy cycles, such as NOPs or bogus program code.

The idea of masking processed data was initally proposed by Chari et al. [7]. They suggested a secret sharing scheme where each bit of the original computed data is divided probabilistically into two or more shares such that any subset of shares is statistically independent of the bit being processed and yields no information about the bit. Goubin et al. proposed a first masking technique for the DES algorithm based on two independent shares in [15]. They describe several methods how to mask intermediate data with an x-or mask generated by a random number generator. In general, masking non-linear functions, such as substitution boxes, is more difficult than linear functions, because it is more difficult to unmask the output data. Unmasking of linear functions can easily be done by applying the x-or mask itself, or the additive or multiplicative inverse. Unmasking of non-linear functions is more difficult and s-boxes usually have to be precomputed for a particular mask.

However, secret sharing methods can be defeated by so-called higher order DPA attacks [23]: two or more points in the side channel curve are analyzed and checked for a correlation of their joint consumption and the unmasked data bit. In order to successfully conduct a higher order DPA an adversary needs to know the exact points of time of all the shares and Chari et al. have shown that the complexity of higher order DPA attacks grows exponentially with the number of shares. A disadvantage of the masking countermeasures is the fact that the performance of an implementation drastically decreases: the number of instructions is roughly doubled, the code size increases and non-linear functions such as s-boxes have to be precomputed and stored in RAM, which is generally sparse in embedded systems. Another disadvantage is the fact that masks corresponding to different group operations have to be converted into each other without revealing the unmasked data. In [5] it is shown that these conversion algorithms eventally do not resist single-order DPA attacks. In [14] DPA-resistant conversion algorithms are presented; however, their performance tends to be a major bottleneck: while a conversion from boolean to arithmetic masking requires seven operations, a conversion from arithmetic to boolean masking requires $5k + 5$ operations, with k being the word size in bits, e.g. $k = 8$. Recently, in [9], Coron et al. proposed a faster algorithm for arithmetic to boolean conversion based on precomputed look-up tables. Depending on the table sizes their approach reduces the number of operations by up to 50 percent.

The second class of software countermeasures aims at the desynchronization of side channel traces by insertion of random dummy cycles or loops of

bogus instructions with a random number of repetitions. There have not been many publications about this class of countermeasures; however, in [6] and [1] it has been shown that bogus instructions can be easily removed by either performing a DPA with an increased number of measurements (e.g. up to several thousands) or by applying signal processing algorithms in order to realign the traces. As a conclusion, desynchronization countermeasures should not be regarded as secure and should only be implemented in combination with other hard- and software countermeasures.

Hardware Countermeasures

Up to now only very few papers have discussed hardware countermeasures, especially since smart card manufacturers do not like to reveal details about their particular developments. To the author's knowledge, there exist three classes of hardware countermeasures against side channel attacks: noise generators, transistor logic families with a constant power consumption and random wait states and internal clock variations. Noise generators add an additional noise part to the overall power consumption. It is obvious that this countermeasure can be easily defeated by computing the mean of several side channel traces corresponding to the same plain text encryption.

The second class is based on the idea of balancing all internal binary transitions, i.e. for every transistor which switches from a logical one to a logical zero, there exists a complementary transistor which switches from a logical zero to a logical one and vice versa. A popular representative of this countermeasure is the dual-rail logic, which also helps to counteract fault attacks [26]. However, it should be noted that such approaches are very difficult to achieve in practice and even slight differences in power consumption may result in a successful DPA attack. Moreover, it is not known whether hardware countermeasures, such as dual-rail gates, can resist EM attacks. Finally, it should also be noted that dual-rail logic results in roughly twice the chip size.

The third class of hardware countermeasures aims at a desynchronization of repeated side channel measurements by invoking random wait states or varying the internal clock frequency. However, as aforementioned, these countermeasures can be easily defeated using realignment algorithms, especially if random wait states result in a distinct power signature. Variations of the internal clock frequency can be easily filtered out with an increased number of measurements, because the random time offsets of significant points within an instruction feature a gaussian distribution.

7 Results and Conclusions

In this publication we review four popular classes of side channel attacks: simple power analysis (SPA), differential power analysis (DPA), internal collision attacks and template attacks. SPA represents the most crude approach

to break implementations of cryptographic algorithms. Despite its name SPA requires that an adversary has a deep understanding of both the attacked hardware and implementation, i.e. the adversary needs to build a precise model of the target's current consumption for the specific instructions observed. This means that the adversary needs the same device to generate these models or a sample device whose behavior is very close to the one under attack. Besides that a deep understanding of the timing behavior, i.e. which instruction is executed at what point of time, is required. DPA overcomes these difficulties; therefore, DPA is generally regarded as the most dangerous side channel attack. Two further attacks, internal collision attacks and template attacks, have the advantage of requiring fewer measurements than DPA in order to determine the secret key. Finally, a brief overview of generic countermeasures against side channel attacks is given. In the past, almost all countermeasures have been broken in the scientific literature, especially if an adversary knows some details about the implementation or target hardware. It must be stressed that countermeasures should never be implemented alone, but always as a combination in order to thwart side channel attacks.

References

1. D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side – Channel(s). In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*. Springer-Verlag, 2002.
2. M. Aigner and E. Oswald. Power Analysis Tutorial. `www.iaik.tugraz.at/aboutus/people/oswald/papers/dpa_tutorial.pdf`. Seminar paper.
3. M. Briceno, I. Goldberg, and D. Wagner. An Implementation of the GSM A3A8 algorithm, 1998. `www.scard.org/gsm/a3a8.txt`.
4. M. Briceno, I. Goldberg, and D. Wagner. GSM cloning, 1998. `www.isaac.cs.berkely.edu/isaac/gsm--faq.html`.
5. C. Clavier and J.-S. Coron. On Boolean and Arithmetic Masking against Differential Power Analysis. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume LNCS 1965, pages 231 – 237. Springer-Verlag, 2000.
6. C. Clavier, J.S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume LNCS 1965, pages 252–263. Springer-Verlag, 2000.
7. S. Chari, C. S. Jutla, J. R. Rao, , and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Advances in Cryptology – CRYPTO '99*, volume LNCS 1666, pages 398 – 412. Springer-Verlag, August 1999.
8. S. Chari, J.R. Rao, and P. Rohatgi. Template Attacks. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, pages 13–28. Springer-Verlag, 2002.
9. J.-S. Coron and A. Tchulkine. A New Algorithm for Switching from Arithmetic to Boolean Masking. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2003*, pages 89–97. Springer-Verlag, 2003.

10. M. Davio, Y. Desmedt, and J.-J. Quisquater. Propagation Characteristics of the DES. In *Advances in Cryptology – CRYPTO '84*, pages 62–74. Springer-Verlag, 1984.
11. J.-F. Dhem, F. Koene, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.L. Willems. A practical implementation of the timing attack. UCL Crypto Group Technical Report Series CG-1998/1, Universiteé catholique de Louvain (UCL), Place du Levant, 3 B-1348 Louvain-la-Neuve, Belgium, 1998.
12. H. Dobbertin. RIPEMD with two-round compress function is not collision-free. *Journal of Cryptology*, 10:51–68, 1997.
13. H. Dobbertin. Cryptanalysis of md4. *Journal of Cryptology*, 11:253–271, 1998.
14. L. Goubin. A Sound Method for Switching between Boolean and Arithmetic Masking. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, pages 3 – 15. Springer-Verlag, 2001.
15. L. Goubin and J. Patarin. DES and differential power analysis: the duplication method. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 1999*, volume LNCS 1717, pages 158–172. Springer-Verlag, 1999.
16. Technical Information – GSM System Security Study, 1998. `jya.com/gsm061088.htm`.
17. H. Heys. A Timing Attack on RC5. In *Selected Areas of Cryptography – SAC '98*. Springer-Verlag, 1998.
18. P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. `www.cryptography.com/dpa/technical`, 1998. Manuscript, Cryptography Research, Inc.
19. P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology – CRYPTO '99*, volume LNCS 1666, pages 388–397. Springer-Verlag, 1999.
20. P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology – CRYPTO '96*, volume LNCS 1666, pages 104–113. Springer-Verlag, 1996.
21. K. Lemke, K. Schramm, and C. Paar. DPA on n-bit sized Boolean and Arithmetic Operations and its Application to IDEA, RC6 and the HMAC-Construction. In *Cryptographic Hardware and Embedded Systems – CHES '04*. Springer-Verlag, August 2004.
22. T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.
23. T. S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume LNCS 1965, pages 238 – 251. Springer-Verlag, 2000.
24. R. Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smart Cards. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume LNCS 1965, pages 78 – 92. Springer-Verlag, 2000.
25. NIST FIPS PUB 180-1. *Secure Hash Standard*. Federal Information Processing Standards, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., April 1995.

26. S. P. Skorobogatov and R. J. Anderson. Optical Fault Induction Attacks. In *Cryptographic Hardware and Embedded Systems – CHES '02*, pages 2–12. Springer-Verlag, 2002.
27. Werner Schindler. A timing attack against rsa with the chinese remainder theorem. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *LNCS*, pages 109–124. Springer-Verlag, 2000.
28. Werner Schindler. A combined timing and power attack. In Pascal Paillier David Naccache, editor, *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002*, volume 2274 of *LNCS*, pages 263–279. Springer-Verlag, 2002.
29. K. Schramm. DES Sidechannel Collision Attacks On Smartcard Implementations. Master thesis: `www.crypto.rub.de`, August 2002. University of Bochum, Germany.
30. Werner Schindler, Francois Koene, and Jean-Jacques Quisquater. Unleashing the full power of timing attack. UCL Crypto Group Technical Report Series CG-2001/3, Universiteé catholique de Louvain (UCL), Place du Levant, 3 B-1348 Louvain-la-Neuve, Belgium, 2001.
31. K. Schramm, G. Leander, and P. Felke. A Collision-Attack on AES Combining Side Channel- and Differential-Attack. In *Cryptographic Hardware and Embedded Systems – CHES '04*. Springer-Verlag, August 2004.
32. Werner Schindler and Colin D. Walter. More detail for a combined timing and power attack against implementations of rsa. In K. G. Paterson, editor, *Cryptography and Coding 2003*, volume 2898 of *LNCS*, pages 245–263. Springer-Verlag, 2003.
33. K. Schramm, T. Wollinger, and C. Paar. A New Class of Collision Attacks and its Application to DES. In Thomas Johansson, editor, *Fast Software Encryption – FSE '03*, volume LNCS 2887, pages 206 – 222. Springer-Verlag, February 2003.
34. Hervé Ledig, Frédéric Muller, Frédéric Valette. Enhancing Collision Attacks. In *Cryptographic Hardware and Embedded Systems – CHES '04*. Springer-Verlag, August 2004.
35. C. D. Walter. Montgomery's Multiplication Technique: How to Make It Smaller and Faster. In *Cryptographic Hardware and Embedded Systems – CHES '99*, volume LNCS 1717, pages 80–93. Springer-Verlag, 1999.
36. N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley Publishing Company, 1993.
37. A. Wiemers. Partial Collision Search by Side Channel Analysis. Presentation at the Workshop: Smartcards and Side Channel Attacks, January 2003. Horst Görtz Institute, Bochum, Germany.

Embedded Security: Physical Protection against Tampering Attacks

Kerstin Lemke

Horst Görtz Institute for IT Security
Ruhr-Universität Bochum
44780 Bochum, Germany
lemke@crypto.rub.de

Summary. Once an adversary gains physical access to a cryptographic device itself, the security of the device strongly depends on its construction implemented in hardware and software. This contribution aims to review the main approaches towards physical security.

Keywords: physical security, tamper resistance, tamper response, tamper evidence

1 Introduction

Physical security becomes essential if a security module is directly accessed, especially in an unprotected environment.

In automotive applications it has to be assumed that each party involved (car owner, workshops, control personnel) is (in principle) able to get physical access to each security component of the vehicle. Certainly, some special tools (as diagnostic interfaces) and design specifications are helpful which are supposed to be distributed to a limited group only. However, the trade of these tools and information can hardly be controlled [16], as for example, independent workshops cannot be excluded from the distribution.

In the automotive industry threats often deal with the modification and substitution of components. The use of cryptographic modules for automotive components is still at the beginning – compared to the efforts of banking associations and national security agencies. Probably many important results towards physical security are still classified and not publicly available. However, there are a number of microelectronics companies that have been developing security modules and hence have gained knowledge concerning how to protect their products against tampering attacks. The automotive industry should benefit from the experience.

The integration of physical security aspects into automotive specifications is a growing issue. The EU directive [13] which regulates the requirements of the digital tachograph system addresses physical security requirements. Besides that, physical security is an issue at the implementation of electronic immobilizers which, for example, should be protected against cloning. Electronic road pricing and digital rights management are further applications which need physical secure components in the end-user environment.

This contribution aims to review the standard requirements for physical security with respect of the security aspects in the automotive industry.

2 The General Model

Cryptographic modules are designed to provide IT security services, e.g. integrity and confidentiality of application data. For the corresponding security mechanisms cryptographic keys are needed which have to be protected themselves against unauthorized disclosure and modification.

We make use of the concept of the "Cryptographic Boundary" [3]. The relevant security parts to be protected are all inside the cryptographic boundary which includes the processing hardware, data and program memories as well as other critical components as a physical Random Number Generator (RNG) or a Real Time Clock (RTC) (see Fig. 1). There are external interfaces to the cryptographic boundary, e.g, for data communication and power supply. These lines are generally untrusted as they can be accessed externally. A special case is a contactless radio frequency (RF) interface: both the data and power transfer is done by the RF field.

2.1 Operational Environment

The operational environment of a cryptographic module can be diverse: ranging from the security server located at the headquarter of a banking association (or a vehicle manufacturer) to security tokens which are handed over to the end user. While it can be assumed that the central security servers are protected by effective *environmental security measures* (e.g. guards, alarm systems and special organizational measures), these measures cannot be enforced in the case of the untrusted end user. If security tokens cannot be protected by the environment, they have to be constructed to protect themselves.

We distinguish *protected* environments, *periodically controlled* and *non-protected* environments. A periodically controlled environment is in an intermediate range. An example would be the random control of the digital tachograph components that are fitted in trucks, whether they have been tampered with or not.

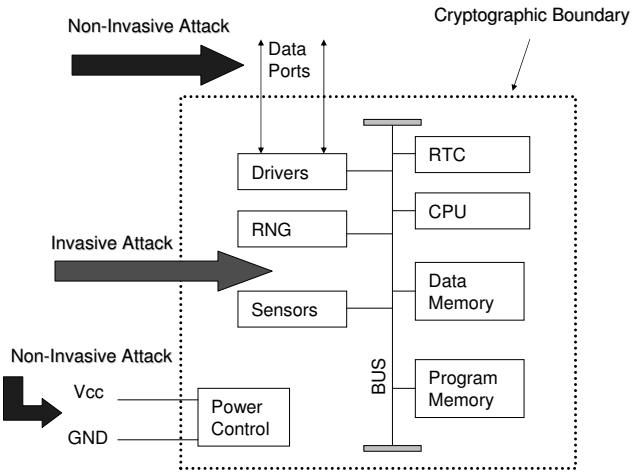

Fig. 1. Components of a computer system that are enclosed in a cryptographic boundary

2.2 Attack Scenarios

In reference [1] five attack scenarios which indicate the main areas of concern are defined: penetration, monitoring, manipulation, modification and substitution.

Penetration: Penetration is an active, invasive attack against the cryptographic module. This includes breaking into the cryptographic boundary of the module. The aim is to intercept data at the internal communication lines or to read out the memory in order to determine the secret keys stored inside the security module.

Monitoring: Monitoring is a passive, non-invasive attack which leaves the cryptographic boundary intact. This class of attack makes use of the inherent leakage of the cryptographic module, e.g. by measuring the electromagnetic emanation. Monitoring by capturing electromagnetic emanation of the cryptographic device (the US military calls this TEMPEST) and "Side Channel Cryptanalysis" are prominent passive attacks based on monitoring. For a detailed description of "Side Channel Cryptanalysis" refer to [14].

Manipulation: Manipulation is a passive, non-invasive attack which leaves the cryptographic boundary intact. The attacks aims to obtain a service in an unintended manner [1], mainly at the logical interface. Manipulating attacks may also include anomalous environmental conditions. For instance, the cryptographic module might be operated under extreme operating conditions, e.g., with short-time glitches in the power supply and at an extreme temperature.

Modification: Modification is an active, invasive attack. This includes breaking into the cryptographic boundary of the module. Unlike penetration

attacks, the aim is to modify internal connections or the internal memories used.

Substitution: Substitution includes the removal of the cryptographic module, which is then substituted by an emulating device with a modified implementation of security functions. The cryptographic boundary is not of primary interest in this attack. Note that the removed module can be used for a comprehensive analysis of the internal construction.

2.3 Physical Security Objectives

There are two different objectives which have to be regarded in physical security.

The first one aims to definitively prevent the disclosure and/or modification of the internal data (e.g. cryptographic keys and application data). For its realization, *tamper-resistant* and *tamper-responsive* measures are implemented. Tamper-responsive measures lead to the zeroization of the cryptographic keys once an attack is detected. Tamper resistance implies that the cryptographic module is able to avert all attacks even without any active reaction.

Another approach focuses on the question whether or not a cryptographic module has been tampered with. For this, *tamper-evident* characteristics are needed. Note that tamper evidence cannot prevent breaking into the cryptographic boundary nor the disclosure of internal data of the cryptographic module. Moreover, the use of a tamper-evident scheme requires a control authority that regularly and carefully inspects the cryptographic module.

According to [1] the terms are defined as follows:

Tamper-evident characteristic: A characteristic that provides evidence that an attack has been attempted.

Tamper-resistant characteristic: A characteristic that provides passive physical protection against an attack.

Tamper-responsive characteristic: A characteristic that provides an active response to the detection of an attack, thereby preventing its success.

2.4 FIPS 140-2: Security Requirements

Among other sets of requirements (e.g. some smart card IC protection profiles used by Common Criteria evaluations [5] and [6]) the FIPS 140 security requirements give an insight into the implementation of secure computer systems for the use in unprotected areas. FIPS (Federal Information Processing Standards) are developed under the National Institute of Standards and Technology (NIST), for use by US federal government departments.

For the evaluation of cryptographic modules the National Institute of Standards and Technology (NIST) in the US published the standard FIPS 140 [4] in 1994. It contains security requirements for cryptographic modules. In 2001, FIPS 140-2 [3] superseded the previous standard FIPS 140-1.

The FIPS 140-2 standard defines four security levels ranging from a low security level to the definition of highly resistant cryptographic modules. The evaluation aspects cover eleven requirement areas:

a. Cryptographic Module Specification,
b. Cryptographic Module Ports and Interfaces,
c. Roles, Services, and Authentication,
d. Finite State Model,
e. Physical Security,
f. Operational Environment,
g. Cryptographic Key Management,
h. Electromagnetic Interference / Electromagnetic Compatibility,
i. Self-Tests,
j. Design Assurance, and
k. Mitigation of Other Attacks.

The requirements in each area are detailed and even go down to the implementation level. Physical security is only one aspect among them. The majority of areas deal with logical functions to be implemented. Other aspects cover the quality of the design documentation. "Mitigation of Other Attacks" is an optional area without concrete test procedures: this area deals with side channel based attacks as "Power Analysis", "Timing Analysis", "Fault Induction" and "TEMPEST".

Regarding to "Physical Security" FIPS 140-2 distinguishes the embodiments

- Single-chip cryptographic module,
- Multiple-chip embedded cryptographic module, and
- Multiple-chip standalone cryptographic module.

The requirements for the physical security increase from level 1 (no special protections) towards level 4 (control of environmental temperature and voltage, single chip: "hard opaque removal-resistant coating", multiple-chip: "tamper detetection envelope with tamper response and zeroization circuitry"). Level 2 and level 3 provide tamper-evident measures; level 3 includes an automatic zeroization when the maintenance (privileged) access interface is entered.

3 Tamper Evidence

Tamper evidence requires an observer (or control personnel) who periodically randomly and carefully inspects the cryptographic device whether a tamper attempt has been occurred.

"Periodically randomly" implies the inspections are conducted in a random, non-predictable way (e.g. street controls in the digital tachograph system). Between inspections, the cryptographic module may reside in a surveilled public

area or in a non-protected environment. In the first case it may be assumed that a potential attacker can be identified afterwards using, e.g., video streaming data; in the second scenario the operator and the corresponding company are responsible for compliance with the legal regulations. Randomness of inspections is important to avoid a regular replacement with an emulating device between the controls.

"Carefully" implies that the observer correctly identifies indications towards tamper attempts and carries out inspections in detail, according to the rules.

Note once again that tamper evidence cannot prevent the disclosure of internal confidential information as well as cryptographic keys. Applications that rely on secrecy of cryptographic keys will never rely on tamper-evident measures if the cryptographic module is not permanently protected.

3.1 Technical Solutions

Typical tamper-evident characteristics include security seals (including special inscriptions and holograms), special covers and enclosures. It is important to note that

- the removal of these items should be sufficiently difficult and leave remaining traces that can be recognized by trained inspection personnel,
- the items should include special characteristics which are not commercially available,
- the faking of these items is sufficiently difficult and can be recognized by trained control personnel, and
- the items are controlled during manufacture and delivery.

Some solutions may include "obscure", i.e., uncommon approaches that are not obvious for an attacker. The combination of such efforts as well as the enclosure of detailed information on its mechanical and chemical construction may achieve an acceptable security level.

More solutions are found in [2], as there are brittle packages and specially prepared surfaces such as "Crazed Aluminium", "Polished Packages" and "Bleeding Paint".

Critical points of an overall enclosure are the lines for data communication and power supply, if any. In cryptographic modules such as contactless smart cards even these lines are omitted.

4 Tamper Response

Tamper response requires that the cryptographic module detects any intrusion attempts at the cryptographic boundary. The cryptographic boundary has to be permanently supervised.

Besides invasive attacks there might be some critical operational conditions which can lead to unforeseen events, such as tampering with the power lines and the environmental temperature. For their detection, special sensors are integrated inside the cryptographic module which permanently supervises the operating conditions.

One general requirement for tamper response is that an internal power supply must be available to detect and react to tamper attempts. Note that smart cards and similar tokens are not equipped with an internal power supply and therefore tamper response measures cannot always be guaranteed.

Another general requirement is that security-sensitive information has to be zeroized as fast as possible, so that the zeroization cannot be stopped by the attacker. Therefore, critical data to be zeroized has to be stored in a RAM-based memory.

4.1 Technical Solutions

Securing the Cryptographic Boundary

The best solution is an active shield at the cryptographic boundary which is implemented by flexible printed circuit sensors [2]. The flexible printed circuit sensors include a mesh of conducting wires which are printed as close as possible to each other. The connection of two near by wires causes short-circuits which can be detected. The wires may be made of silk-screened conductive paste which provide a high resistance and are difficult to attach to. The mesh is embedded into a potting material. Attempts to remove the potting material impacts on the overall resistance of the wires, which is permanently measured and observed. Especially, chemical attacks result in both dissolving the potting and the insulation between the wires.

Any break-in will result in ignition of the zeroization circuitry.

Environmental Sensors

For the control of environmental conditions, the cryptographic module is equipped with temperature and voltage sensors, and if applicable with additional sensors (e.g. for motion).

Whereas temperature changes expand slowly, the reaction time is not as critical as in the case of tampering attempts on the external voltage line and which demand an instant reaction.

One special critical case occurs if the internal power supply is no longer capable to activate the zeroization circuitry. To avoid this case, the zeroization circuitry has to be fired before the critical state of the internal power supply is reached.

Another requirement might be that the cryptographic module detects any removal attempts. For these kinds of attacks, GPS devices and movement sensors are appropriate.

Zeroization Circuitry and Data Remanence

The speed of the zeroization is cruical as an interruption would leave the cryptographic module in an intermediate state which might contain sensitive data. The preferred solution is a hardware-based implementation.

Another question deals with the fact that just a power-down of the SRAM data storage might be not sufficient due to data remanences caused by "burn-in" of the data over a long time period (see [11]). Typically, the RAM has to be actively overwritten multiple times. One practical solution to minimize these effects is to periodically update the representation of the data stored in SRAM, e.g. by using an encryption key which is periodically changed internally. In [11] and [2] an alternative destructive approach for very sensitive applications is proposed: the exposure of the cryptographic device to high temperatures.

Preventing Monitoring Attacks

Monitoring attacks at the external interfaces cannot be detected by the cryptographic module. It is therefore necessary that these kinds of attacks based on electromagnetic emanation are made as difficult as possible.

One possible approach is that of [12] who proposed to switch capacitances between the internal power supply and the external power lines so that data-dependent signals at the external lines are minimized.

For the electromagnetic emanation, special shielding cases might be a certain solution.

5 Tamper Resistance

Typically, cryptographic modules claiming tamper resistance cannot actively detect tamper attempts in all cases. Prominent examples are smart cards that are supplied with voltage and clock externally. If the smart card is not powered on, penetration and modification attempts cannot be detected.

The internal construction of tamper-resistant modules has to withstand physical attacks.

Note that a removal of a tamper-resistant module can generally not be prevented by the module itself.

5.1 Technical Solutions

The requirements to be fulfilled for tamper-resistant ICs are set down in the protection profiles [6] and [5]. Appropriate technical solutions can be derived from these requirements.

Special Design Characteristics

It is aimed to counteract reverse-engineering as much as possible. Some typical measures in the internal construction include the encryption of internal bus lines and memories which contain critical persistent data. Moreover, the layout should contain special characteristics, such as the scrambling of bus lines and memories as well as special logic styles.

Another important fact is the shrinking of structure widths in the semiconductor industry. Upcoming transistor technology is based on 90 nm gate widths. This technology demands specialized equipment, such as FIB (focused ion beam) workstations.

During power-on simple modifications using microprobing workstations can be averted by the use of an active shield on top of the metal layers.

It is further important to note that test features used during manufacturing will no longer be available in the operating environment. Such requirements have to be specified as part of the life cycle of the product.

During start-up of the tamper-resistant module it is recommended to include self tests which verify the integrity of critical internal data. A special focus lies on the internal hardware-based random generator, which might be needed for special countermeasures. A destruction of the random number generator has to be detected during start-up and operation.

Preventing Fault Attacks

During operation the cryptographic module might be exposed to environmental conditions that are outside of the secure operation range. Critical operating conditions have to be recognized immediately. Therefore, the tamper-resistant module should be equipped with sensors for temperature as well as for the supervision of the voltage and clock supply. Operation outside the secure range of parameters has to be prevented, and a secure state has to be entered, e.g. by enforcing a reset condition.

Actually, state-of-the-art fault analysis techniques make use of light injection and cause photoelectric effects in de-packaged integrated circuits. Optical fault induction allows for a great spatial precision: it is possible to target one SRAM cell. A survey on methods of fault induction and their effects can be found in [15]. Faults can be detected by implementing error detection codes or by repeating the computation of security-relevant parts. Physical countermeasures include light sensors, an improved shielding of security-relevant areas, and special logic styles such as “dual-rail” logic.

Preventing Monitoring Attacks

Monitoring attacks at the external galvanic interfaces as well as at internal lines (after some successful internal modification during power-down) cannot

be detected by the cryptographic module. This applies also to electromagnetic emanation.

It is therefore necessary that these kinds of attacks are made as difficult as possible, i.e. the monitored source should reveal as little information about undergoing processes as possible. Special logic styles such as "dual-rail" logic and asynchronous circuits are an appropriate solution. If this is not sufficient, additional measures should be implemented by software, especially for the implementation of cryptographic algorithms. We refer to [14] for possible solutions. In general, the most secure results can be achieved when different countermeasures (e.g. soft- and hardware based) are combined. However, it should be noted that countermeasures usually implicate a degradation in performance, either in terms of chip size in the case of hardware countermeasures or in terms of efficiency and code size in the case of software countermeasures.

6 Conclusion

In this contribution the basic requirements for physical security are summarized. Some appropriate solutions are presented.

Finally, it should be emphasized that the strength of physical security measures is limited in principle. Detailed knowledge on the implementation combined with advanced analysing equipment and sufficient attacking time might still overcome some sophisticated solutions. The strength of mechanisms according to formal security evaluation criteria (such as the Common Criteria and the ITSEC) is assessed basically according to specific preconditions that are necessary for an attack to be mounted sucessfully, namely the knowledge and expertise of an attacker, the equipment, the time and the number of samples that are needed.

References

1. ISO 13491-1, Banking – Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods, First edition 1998-06-15.
2. Steve H. Weingart. *Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses*, in Cryptographic Hardware and Embedded Systems – CHES 2000, LNCS 1965, Springer, 2000.
3. FIPS PUB 140-2, Security Requirements for Cryptographic Modules, National Institute of Standards and Technology, 2002, available at `csrc.nist.gov/cryptval`.
4. FIPS PUB 140-1, Security Requirements for Cryptographic Modules, National Institute of Standards and Technology, 1994, available at `csrc.nist.gov/cryptval`.
5. BSI-PP-0002: Smartcard IC Platform Protection Profile, 1.0, available at `www.bsi.bund.de/cc/pplist/ssvgpp01.pdf`.

6. PP/9806: Smartcard Integrated Circuit Protection Profile v2.0, available at `www.ssi.gouv.fr/site_documents/PP/PP9806.pdf`.
7. Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, Inc., 2001.
8. Sean W. Smith, Steve Weingart. *Building a High-Performance, Programmable Secure Coprocessor*, IBM T. J. Watson Research Center, Revision of October 16, 1998, available at `www.research.ibm.com/secure_systems_department/projects/scop/papers/arch.pdf`.
9. Ross Anderson, Markus Kuhn. *Tamper Resistance – a Cautionary Note*, The Second USENIX Workshop on Electronic Commerce Proceedings, USENIX Association, Oakland, California, November 18–21, 1996, pp 1–11, ISBN 1-880446-83-9, available at `www.cl.cam.ac.uk/~mgk25/tamper.pdf`.
10. Oliver Kömmerling, Markus G. Kuhn. *Design Principles for Tamper-Resistant Smartcard Processors*, USENIX Workshop on Smartcard Technology proceedings, Chicago, Illinois, USA, May 10–11, 1999, available at `www.cl.cam.ac.uk/~mgk25/sc99-tamper.pdf`.
11. Peter Gutmann. *Secure Deletion of Data from Magnetic and Solid-State Memory*, Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996, pp 77-90, available at `www.usenix.org/publications/library/proceedings/sec96/full_papers/gutmann/`.
12. Adi Shamir. *Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies*, Cryptographic Hardware and Embedded Systems – CHES 2000, Springer, Lecture Notes in Computer Science, Vol. 1965, Berlin (2000), 71–77.
13. Commission Regulation (EC) No 1360/2002 of 13 June 2002 adapting for the seventh time to technical progress Council Regulation (EEC) No 3821/85 on recording equipment in road transport, Annex 1 B, Requirements for Construction, Testing, Installation and Inspection.
14. Kai Schramm, Kerstin Lemke, Christof Paar. *Embedded Cryptography: Side-Channel Attacks*. This book.
15. Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, Claire Whelan. *The Sorcerer's Apprenctice's Guide to Fault Attacks*, available at `eprint.iacr.org/2004/100`.
16. 'Die neue Strategie der Autodiebe', Frankfurter Allgemeine Zeitung, 17.02.2004, Nr. 40 / Seite T1.

Part IV

Business Aspects of IT Systems in Cars

Automotive Digital Rights Management Systems

Marko Wolf[1], André Weimerskirch[2], and Christof Paar[1,2]

[1] Horst Görtz Institute (HGI) for IT Security,
Ruhr University of Bochum, Germany
{mwolf,cpaar}@crypto.rub.de

[2] escrypt GmbH, Bochum, Germany
{aweimerskirch,cpaar}@escrypt.com

Summary. Increasingly powerful embedded computers and multimedia technologies enter the automotive world on a grand scale, various new opportunities as well as some important challenges will become reality. Location-based up-to-date navigation and infrastructure data, movies, music, entertainment software or additional vehicular features and setups on demand will become ubiquitous within the next automotive generation, but have to be reliably protected against unauthorized usage or manipulation. Whereas most copy protection mechanisms solely try to prevent illegal copying of digital content, systems for digital rights management (DRMS) allow rights holders to implement and enforce a detailed rights model which exactly controls the utilization rights. Moreover, protecting vehicular data and software against unauthorized manipulation means for both drivers and car manufacturer an enormous safety advantage. But in comparison to common application domain of DRMS in the world of personal computers, the application in the automotive domain involves several crucial restrictions and specific requirements.

1 Introduction

Together with today's computer and multimedia technologies, protected digital content enter also the automotive world on a grand scale. Various new opportunities as well as some important challenges will become reality. Location-based up-to-date navigation and infrastructure data, movies, music, entertainment software or additional vehicular features and setups on demand will become ubiquitous within the next automotive generation, but have to be reliably protected against unauthorized usage or manipulation. As today's computer technology significantly simplifies the creation, copying and utilization of intellectual properties, it simplifies the circumvention of one's legitimate rights as well. Thus, possible right holders are willing to market their digital content only if they can rely on an effective protection of their digital commodities. On the other hand, potential users will use the new possibilities

and are finally willing to pay for digital content only if privacy, ease of use, interoperability and availability can be provided reliably. Whereas most copy protection mechanisms solely try to prevent illegal copies of digital content, systems for digital rights management allow rights holders to implement and enforce a detailed rights model which exactly controls the utilization rights. In addition to copy protection and flat fee billing, digital rights management systems (DRMS) are also capable of offering almost arbitrary combinations of usage-based accounting methods (time-based, quantity-based, device-based, etc.).

2 New Applications for Automotive DRMS

Protecting vehicular data and software against unauthorized manipulations induces both for drivers and car manufacturer an enormous safety advantage. It prevents, for instance today's common frauds such as mileage counter manipulation, unauthorized chip tuning, or tachometer spoofing. Securing particular important vehicular data against unauthorized access and manipulation, i.e. enforcing the respective access policies, is mandatory for applications such as the digital tachograph, electronic license plate, all kinds of drive-through payments (toll, gas, parking, etc.), compliance with particular warranty and insurance regulations, or legal restrictions on exhaust emissions or engine power. Moreover, any unauthorized, potential faulty software update can jeopardize the safety of the entire vehicle.

2.1 New After-Sales Business Models

Perhaps the most exciting new application in the automotive industry is driven by new after-sales business models. The introduction of a number of new multimedia formats and personalized location-based information services promises a wide, lucrative additional market for the automotive industry. Today already most medium-sized cars are equipped with multimedia-capable on-board computers and radio systems. An upgrading for DRM functionality will enable various new business models for usage-metered and on-demand utilization of digital content, software and even hardware beyond the classical lump-sum model. Some possible examples are provided below.

Time-limited utilization: Up-to-date navigation data may be available on demand for any place in the world (e.g. for a two-week vacation trip in the respective area).

Quantity-limited utilization: Movies, music tracks, or games can be bought for an n-times repeated utilization.

Device-bound utilization: Extra software can be installed solely on a particular device or a particular vehicle. Certain car functions are performed

only via a certain authentication device such as a driver's key, dealer token or personal cellular phone.

Usage-metered utilization: Navigation routes can be charged for their actually used length. Movies or music tracks can be charged for the actual viewing time.

Furthermore, almost arbitrary combinations are possible. For instance, afterwards activated enhanced comfort sensors (e.g. tire air pressure sensor) may be enabled as a free sample for 4 weeks. Clearly, all these business models are only possible when a trusted computing base is available and a DRMS that implements the rights management. Having no such secure anchor, the business model will certainly fail. A popular example here is the pay-TV provider Premiere.

2.2 Upgrade Activation

Since the production of vehicular components moves from various small batches of different, individually adjusted components towards large-scale production of only a small number of standardized parts, today cars consist mostly of the same components. Therefore cars of different equipment categories are distinguished from each other only by the amount of per default activated features. Given that some features are built in already, but locked by default, a DRMS can securely activate (or even deactivate) extra hardware or software components afterwards for an additional charge. Features that would be capable of after-market activation could include special setups for engine, gear or chassis control, enhanced on-board computer and comfort diagnosis functions, additional driving assistance and infotainment capabilities or certain personalization features.

2.3 Secure Flashing

More and more vehicles are equipped with ECUs that use flash memory. Flash memory can be updated in order to download new software versions to the vehicle. For instance, a flawed version can be patched. However, flash memory in ECU provides far more features. Several versions of an ECU can be based on the same hardware platform. The functionality then only differs because of the software. This enables cost-efficient logistics as well as reduced development cost. Clearly, the process of flashing must be protected by cryptographic schemes. Only authentic software must be downloaded to the ECU and often it is desirable that the downloaded software is encrypted. Hence, the ECU must be able to verify and decrypt the downloaded software. Clearly, such protection schemes are only possible if they are based on a trusted hardware base. Otherwise, an adversary might just manipulate the ECU in order to download any software.

2.4 Secure Infotainment

Today's vehicles have become multimedia centers. Besides music there are video capabilities as well as navigation capabilities available. Mobile music players, mobile navigation systems, personal digital assistants (PDAs), as well as cell phones are attached to the car's central computing unit. Multimedia files stored on a mobile player that are protected by a DRMS should be transferred to the car's multimedia system to play them. However, the vehicle's head unit then needs to support the same DRMS. Hence, a DRMS in the vehicle is required. It is advantageous if the car provides a multiple DRMS supporting DRMS of several providers.

2.5 Further Applications

There are several further applications enabled by a trusted computing base:

- Component identification provides a mechanism to avoid vehicle theft and theft of car's components as well as a mechanism against faked spare parts. The electronic immobilizer can be seen as a special instance of component identification. Clearly, a secure platform is required to avoid manipulation of the computing platform to circumvent the scheme.
- Secure data storage provides a mechanism to store operating data securely in a vehicle. An example is a secure tachograph required for trucks. This data must not be altered. Often, data storage must comply with legal requirements. Hence, a secure data storage provided by a secure platform base is required here.
- Car-to-car communication and car-to-infrastructure communication might be wide spread future applications. For instance, in the future vehicles might establish communication channels to each other in order to exchange warning and information messages, or consider the exchange of messages about free parking spaces.

Furthermore, in the future electronic license plates instead of traditional ones might be used that broadcast the license plate number. Clearly, the communication as well as the stored data must be resistant to manipulation, and accessing the involved data must be regulated by a DRMS.

3 DRMS in General

Digital rights management systems enable flexible electronic selling environments which continuously enforce the compliance of the authorizations of the rights holder. Figure 1 shows a general DMRS model. Thus the three core parties within a DRMS are the content provider C, who is either the rights holder itself or acts on their behalf, as well as the respective user/consumer U, who utilizes the contents granted by C. The central DRMS component

manages the utilization requests of U while reliably enforcing the terms of use of C. At the same time, the DRMS verifies the successful gratification of the respective considerations. Beyond that, most DRMS realizations require an additional communication channel for (initial) administrative purposes. The

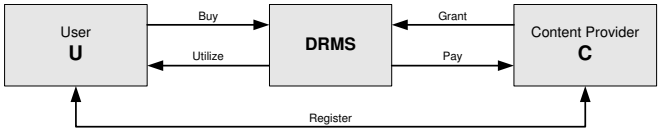

Fig. 1. Basic DRMS model based on [6]

DRMS reference architecture according to [5] consists of three major components: the content server, the license server and the user client. The content server holds the utilizable digital contents and their associated metadata (id, title, author, format, etc.). An encryption-based DRM packager merges inextricably the digital content together with their metadata and a set of rights to protect them for distribution within the scope of a DRMS. Whereas the virtual content is encrypted, associated metadata may remain in plain text, protected only against tampering. Using the integrated accounting system, the license server creates according to user identities and rights descriptions appropriate licenses along with the necessary keys for user authentication and content decryption. Thus such a license includes information about user identity (person, role, license, etc.), a unique identifier as well as the respective parameters of the associated rights model. Components of the DRM reference architecture that reside within the user client are the DRM controller, the rendering application, and the user authentication mechanism. The DRM controller can be an independent piece of software, but can also reside within the rendering application (such as Microsoft's media player) or could be a piece of dedicated hardware. Since DRMS allow rights holders to implement a complex rights model, it is possible to assign a multiplicity of different utilization rights to digital content. Such a rights model defines types of rights and the respective attributes of those rights. According to [7] the four fundamental types of rights are:

- Render rights (e.g. play, view, or print)
- Transport rights (e.g. copy, move, or loan)
- Derivative work rights (e.g. extract, edit, or embed)
- Utility rights (e.g. backup, caching, or integrity checks)

Each of the fundamental rights has particulars attached, called the rights attributes. The three important kinds of rights attributes are:

- Considerations (e.g. money, registration, or membership)
- Extends (e.g. How long? How often? Where?)

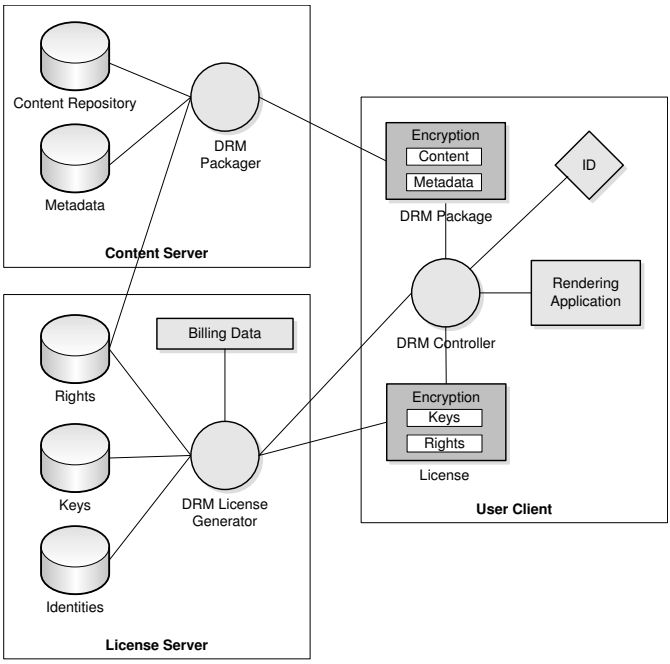

Fig. 2. DRM reference architecture based on [5]

- Types of users (e.g. driver, owner, subscriber, or anonymous)

Hence, if a user requests to access DRM-protected content, the DRM controller gathers the user's identity and asks the license server along with the package identifier for a license that meets the user's demands. If the controller receives an appropriate license and successfully verifies authenticity and integrity of the rendering application, it retrieves the encryption keys from the license, decrypts the content and releases it to the henceforth trusted rendering application. Furthermore, a complete DRMS requires besides the mentioned core components for content protection and utilization also a capable distribution infrastructure, billing system and customer relationship management (CRM). Because only the user client component is subjected to the specific restrictions within the automotive area, we discuss explicitly only that part and give merely a short overview of the required DRMS backend components. Further details can be found in [5, 7, 1, 2].

4 Requirements for Automotive DRMS

In comparison to the common application domain of DRM user clients in the world of personal computers, porting into the automotive domain means

several crucial restrictions and specific requirements. The most important requirements for automotive DRMS are provided below.

Physical Environment: Any electrical component within the vehicular area has to deal with a wide temperature spectrum of up to 100℃, high humidity, and high mechanical load. Moreover, all components have to endure virtually failure-free for the complete life cycle of a vehicle of up to 20 years with only a minimal amount of maintenance.

Embedded Systems: The available computing power and storage space of today's cars differ clearly from that available within the world of standard personal computers. Hence, present PC software modules hardly can be reused within the automotive area and thus we need new software that meets the particular requirements on runtime efficiency and memory extents. Moreover, we will face many architecture-specific constraints.

Cost Efficiency: The overhead induced by the trusted vehicular computing base must be cost-efficient. A solution will only be accepted for the mass market if the financial overhead is very little. Since cost is of such importance, a solution has to involve only minimal hardware supported by software measures.

External Communication: Cars normally have only limited external communication capabilities in both bandwidth and frequency. Therefore, automotive DRMS have to manage license requests, key exchanges and software updates with only minimal external communication and almost fully automated. Most of the DRMS functionality must be applicable even if the external communication is unavailable for an indefinite time.

User Interface and Usability: Whereas computer users are able to deploy various ergonomic input and output devices, most mobile users have to use only limited ergonomic peripheral equipment to control their applications. When mobile users have to perform inevitable interactions, size and complexity of input and output data should be limited to allow fast and smooth handling also in the automotive context (i.e. keypads or rotatory knobs, small-size screens, etc.).

Infrastructure and Interoperability: The necessary key and certificate infrastructure is a specific challenge in the automotive area. The multiplicity of involved parties (manufacturers, suppliers, OEMs, customers, service personnel, content providers, etc.) requires complex and reliable organizational structures. Furthermore, we need a maximum of interoperability with other existing (also non-automotive) DRMS so that customers can easily integrate their existing digital content and familiar devices.

Maintenance and Safety: Cars normally have only limited possibilities for hard- and software maintenance. Therefore, compatibility, reliability and low maintenance effort are mandatory. In particular it is mandatory to have

all needed hard- and software tools easily available during the complete life cycle of the vehicular target platform.

5 Realization of Automotive DRMS

To realize an automotive DRMS we need a mutually trusted vehicular computing base that addresses the security requirements and objectives of all participating parties; i.e. rights holders can rely on an effective protection of their digital commodities while privacy, ease of use, interoperability, availability and fairness are reliably assured on the user's side. We would like to stress that every individual software approach always can be undermined by several hardware manipulations. While the whole hardware is completely under the control of the user and all software fully relies on the underlying hardware layer, it is impossible to implement a mutual trustworthy base with an individual software-based approach. Therefore, we introduce a hardware security module (HSM) that establishes trust in the automotive platform for both user and content provider. An HSM protects all important keys, is able to reliably perform several symmetric and asymmetric cryptographic functions (e.g. 3DES, RSA, ECC, SHA-1 Ĕ) and includes a true random number generator (TRNG) which provides authentic coincidence. Since trust is closely connected with the prevention of hardware manipulations, the HSM has to be especially tamper-resistant [3, 4]. For this, an HSM contains several sensors that monitor, for instance, the incidence of light, temperature, resistance, or clock frequency. Moreover, it may have an extremely dense circuit arrangement and a non-deterministic clock signal. In case one of the online self tests would fail, the HSM would immediately initiate appropriate active countermeasures such as self-deactivation or even self destruction. Hence, a tamper-resistant HSM can be successfully manipulated only at such high cost that exceeds all potential benefits. Based on the HSM, which is connected to the central gateway, we can implement a secure reliable software layer (operating system). This trustworthy layer provides a strict separation from security applications and potentially insecure standard software. Moreover, by means of such a security anchor we are able to utilize and enforce the DRMS throughout the whole vehicle.

Physical Environment: Most physical requirements that apply to DRMS hardware components apply also to other electronic components within a vehicle. Hence we can adopt already available know-how, technologies and precautions almost analogously. In particular, a few smart cards and tamper-resistant processors that meet the characteristic automotive requirements are already available.

Embedded Systems: As we know from our experience even computationally expensive or memory-expensive cryptographic methods can be realized in restricted environments if implemented in a sophisticated manner.

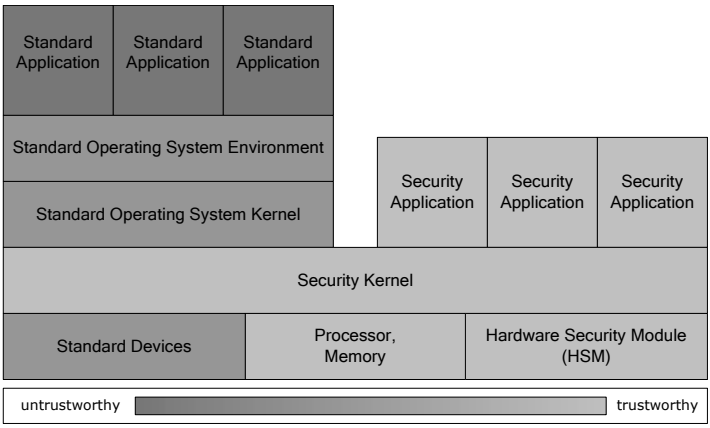

Fig. 3. Trusted platform with a micro kernel based security layer

Especially symmetric cryptographic methods such as AES (Advanced Encryption Standard) or DES (Data Encryption Standard) can be implemented on a small controller processor with only low clock speed. Implementations of asymmetric cryptographic methods such as RSA (according to the inventors Rivest, Shamir and Adleman) and ECC (Elliptic Curve Cryptography) within embedded environments require far more effort to achieve acceptable runtimes. Nevertheless, efficient implementations of cryptographic methods are possible already today even within embedded environments. In particular, up-to-date elliptic curve cryptographic implementations are able to fulfill the specific automotive requirements on runtime and memory efficiency.

Cost-Efficiency: Since automotive DRMS have not entered the mass market so far, realizing cost-efficient automotive DRMS is still a huge challenge. Hence only the application of standardized, interoperable hard- and software components will enable cost-efficient mass production and hence meet the high demands on cost-efficiency.

External Communication: If we delimitate online license purchases, we need external communication only if we bring in new, still unlicensed software or media. At least within Europe, the GSM (Global System for Mobile Communication) coverage suffices to realize any automotive business model associated with protected content. The upcoming area-wide UMTS (Universal Mobile Telecommunication System) mobile transmission standard will enhance further DRMS capabilities such as video streaming, online software applications, and online access to any data.

User interface and Usability: Due to the fixed automotive utilization context and a vehicle-bound unique identity (VID) that avoids repeated user authentication, inevitable interactions with the DRMS could be minimized.

The few remaining inevitable input and output control procedures may be accomplished by already existing user interfaces that belong to common telematics and multimedia equipment at this stage. Moreover, the VID provides privacy and anonymity for all transactions that imply only that anonymous VID.

Infrastructure and Interoperability: The DRMS distribution and administration infrastructure has to be provided either by the car manufacturer or (even better) by an external overall service provider. Unfortunately, the major content providers so far have not agreed on common standards for the important hardware and software interfaces of their DRMS. Hence, current DRMS providers today use only proprietary, mutual incompatible formats and interfaces. However, only the efficient interoperation between most major DRMS providers including those in the non-automotive area could leverage their DRM systems beyond closed niche markets into a lucrative mass market by the use of standardized formats and interfaces.

Maintenance and Safety: The automotive specific high requirements on compatibility, reliability, and low maintenance of a DRMS can be accomplished by various approved methods in software quality management and software verification as it does apply to any other automotive software, too. However, the fast and continuous technical progress, in comparison with other vehicular software, can be controlled only by occasional software updates and additional precaution measures such as extra strong encryption and appropriate fall-back systems. Ensuring maintainability by provision of all necessary hardware and software tools is mostly regulated by law within the indenturing functional guaranty over the whole vehicular life cycle.

6 Security Aspects and Remaining Challenges

Since vehicular DRMS control and store valuable and legally protected data, they are inherently attractive targets for malicious attacks or manipulations. Besides the car owner, also garage mechanics (mostly on behalf of the car owner) and third parties such as competing manufacturer or further unauthorized persons and institutions may have reasonable intents for attacks on a DRMS. Moreover, in contrast to most common computer networks, the car owner and the garage personnel have full physical access to the whole vehicular network and all physical components of the DRMS. Even if the car owner himself normally has only low theoretical and technical capabilities, garage personnel and some external third parties may have both adequate background knowledge and the appropriate technical equipment for performing an attack. Therefore, as long as a successful attack results in a high gain, we have to consider even particularly expensive and sophisticated attacks on the DRMS hardware and its (cryptographic) system design. In particular, if an attack is of such a kind that the attacker can establish his own business the

worst case scenario occurs. An appropriate system will be designed in such a way that a single compromised device does not give any advantage for attacking another device. An asymmetric crypto scheme implemented by the HSM can secure each device individually so that a successful attack would compromise only that particular device, whereupon the necessary technical and financial expenses would simply exceed the value of all potential benefits. Due to the fact that today's DRMS offer only unilateral security, i.e. they focus mainly on requirements of the rights holders or content providers, and not on those of users such as privacy or backup restrictions, we need a multilateral secure DRMS that is able to enforce the policies defined by the rights holders and users equally. Multilateral secure DRM architectures already exist [6] and enable users to decide independently which personal data (particulars, user profiles, etc.), to whom and on what terms they are willing to provide. Moreover, they assure privacy against unauthorized third parties and are capable of enforcing the policies of rights holders equally. For transport (i.e. transfer into another car) and backup of contents and rights, we need appropriate interfaces to support the application of DRM-capable smart cards or protected online storage sites.

7 Summary

Automotive DRMS are the enabler for various new innovative technologies including new automotive business models and exciting applications. At the same time safety-relevant vehicular components are protected against unauthorized manipulation. The basis for DRMS is a secure hardware anchor represented by a TPM. While for PC hardware platforms the first TPM and DRMS solutions are currently introduced, more work needs to be done for developing a platform suited to the automotive environment. However, we believe the first such platform will be available very soon.

References

1. Eberhard Becker, Willms Buhse, Dirk Guennewig, and Niels Rump. *Digital Rights Management – Technological, Economic, Legal and Political Aspects*, volume 2770 of *LNCS*. Springer, 2003.
2. Joan Feigenbaum. *Digital Rights Management*. Springer, 2003.
3. O. Koemmerling and M. G. Kuhn. Design principles for tamper-resistant smartcard processors. In *USENIX Workshop on Smartcard Technology Proceedings*, Chicago, Illinois, USA, May 1999.
4. Nasir Memon and Gleb Nauvomich. *Preventing piracy, reverse engineering, and tampering*, volume 36, pages 64–71. Computer, USA, 2003.
5. B. Rosenblatt, B. Trippe, and S. Mooney. *Digital Rights Management*. M&T Books, New York, USA, 2002.

6. A.-R. Sadeghi and C. Stueble. Towards multilateral-secure DRM platforms. In *Information Security Practice and Experience Conference (ISPEC)*, Singapore, April 2005.
7. M. Stefik. *Letting Loose the Light: Igniting Commerce in Electronic Publication.* MIT Press, USA, 1997.

Security Risks and Business Opportunities in In-Car Entertainment

Marcus Heitmann

Institute for Security in eBusiness (ISEB)
Department of Economics
Ruhr-Universität Bochum, Germany
marcus.heitmann@volkswagen.de

Summary. In-car entertainment is part of many new business models in the automotive industry. The security of such services and products has only played a minor role in these concepts. However, business opportunities should not be handled separately from security. This contribution sums up some of the most important factors for a successful strategy in in-car entertainment products and services.

Keywords: in-car entertainment, security, billing, pricing, infrastructure

1 Introduction

In recent years two developments have progressed rapidly in the automotive industry. First, there are security technologies like ABS, ESP, airbags and many other developments that have enhanced the safety of cars. Second, there are technologies that have made cars a much more convenient and entertaining place to travel in, like navigation systems, CD changers, mobile communications or even in-car home theater equipment like DVD players.

A lot has happened to in-car entertainment (ICE) since the first radio in a Ford T model in 1922. Blaupunkt introduced with the AS 5 the first European car radio in 1932. In 1968 the Dutch company Philips added a cassette drive, in 1985 the CD player and in the same year the first video systems for cars became available too. The car became a media center in 1993 when (video-enabled) navigation systems, CD changers and mobile telephony were available in one device [9].

Modern ICE is already a success. Not only the basic standards like the integration of mobile phones, CD players or CD changers! Sales of equipment like rear-seat entertainment and navigation systems are constantly on the increase – at least in high priced vehicles.

Nearly every analyst worldwide is convinced that the domain "mobile infotainment" has enormous growth potential. A forecast from Frost & Sullivan

states a possible business volume tripling from 2.9 billion euro in 2004 up to 9.2 billion euro in the year 2010.

The needs for ICE are not the same for all parts of the world. The automotive domain is generally divided into three parts: the so-called triad, consisting of Western Europe, USA and Japan. However, these three regions themselves are not homogeneous: the needs of an Italian customer are different from those in Sweden. In 2002 39 million vehicles or 70% of the world's production have been sold in the triad. Substantial growth can be found only in Eastern Europe and parts of Asia [1].

But the rise of ICE is not only a business opportunity it can also pose a serious threat to all parties involved when security issues are not solved properly. IT security in ICE is not only a simple benefit for the OEM or customer but also a strong and powerful business enabler. Some business cases would even be unthinkable without IT and embedded security.

This contribution will sum up technologies, markets, services, customer needs and security issues in the area of ICE. This requires some knowledge of the technical possibilities and opportunities, which will be discussed in Section 3.

2 Convergence Tendencies and the Market for ICE

In the past decade information technology (IT) became an important part of many products and services. Since then one can see an increasing convergence in the IT, telecommunication and entertainment sectors.[1] An IT company such as Apple can design, produce and sell something like the iPod with huge success, while home entertainment specialist Sony does it the other way around with its Vaio laptop series. The software company Microsoft invests in High Definition Video (HDV), sells home entertainment equipment like the X-Box and develops operating systems for mobile phones which connect the features of a Personal Digital Assistant (PDA) with those of a mobile phone.

The fusion between telecommunication and informatics is called telematics and describes the connection between a (mostly) mobile application and a geographical information system. The best (and probably the first) example is navigation software used in cars, nowadays even PDAs or mobile phones.

In-car entertainment is a good example for converging markets: If there is a mobile phone in a car why not connect it to the navigation system and display traffic information? If there is a navigation system with a monitor in a car, why not using it for entertainment purposes as well?

ICE can be divided into two functional groups: first, plain entertainment, where the medium with the content resides inside the car, like a CD, DVD or a memory stick; second, entertainment where the media are provided from

[1] For a detailed description of the term convergence please see [2].

outside of the car through wireless technologies (i.e. *streaming media*).[2] It is difficult to tell if the latter will become a success in cars but it will definitely get pushed by content providers and owners. For them is a new distribution channel where they possibly can circumvent retailers and dealers. Manufacturers of ICE components and systems see a revenue potential likewise.

Therefore the most important topic is streaming audio and video for ICE right now. While streaming audio is already available, streaming video still faces some technical difficulties due to higher data rates and the need for advanced error correction. Additionally there are several other problems: the quality of service has to be guaranteed, i.e. every service needs its specific and guaranteed bandwidth. Especially streaming video needs a lot of bandwidth, which is difficult to deliver to a moving target without a large antenna or satellite dish.

It is important for engineers and management to understand the nature and drivers of innovations to fulfill the demands and challenges of customers and legislation in the automotive industry. There are three kinds of different innovations in cars (not limited to safety and security) [1]:

a. *Must-have technologies*: These technologies rely on legal requirements or safety issues like airbags, ABS or ESP. They reach a high market penetration sometimes within one product life cycle.[3]
b. *Nice-to-have technologies*: Technologies that are responsible for comforting passengers in a car like air conditioning, power windows, central power locking, etc. Most of the time they need about two product life cycles for a high market penetration.
c. *Niche technologies*: These are technologies that are based on individual enthusiasm like memory functions, massage seats, etc. Sometimes it may take several product life cycles until just a small proportion of the vehicles on the market contain this technology.

The decision, of which technology will become a must-have or nice-to-have remains with the customer and the legislation.[4] There are reliable estimations that smart navigation systems will be widely available in European compact class vehicles by 2008 and external networking by 2014.

---

[2] Streaming media is a technique where it is not necessary to have the complete audio or video file at hand like in a *store-and-forward principle*. The receiver of the media buffers the incoming data stream and starts to play the content. While playing, the device receives the next part of the data which then gets played. This process reiterates until the complete file has been played. Normally the data does not remain in the system – a replay of the data requires starting the transmission again (if possible).

[3] The product life cycle of a vehicle is assumed to be approximately between 7 and 10 years.

[4] Sometimes these decisions are made very fast. A law against the manipulation of the odometer of vehicles in Germany was based on an article in a magazine and took only a few months to pass the corresponding legislative bodies.

Drivers for innovation differ depending on the region [1]. In NAFTA the drivers for innovation are legal requirements, comfort features and the potential to reduce costs. Innovations in Japan and Asia get driven by individual customizing and comfort, while in Europe it is safety and security, comfort, prestige and ecofriendliness.

2.1 Services

The following list of services in the mobile domain shows some of the expected potentials (not reduced to ICE):[5]

- Financial services (payment via mobile devices, financial transactions, etc.)
- Information services (weather reports, news, sports, based on personalized profiles, etc.)
- Entertainment services (games, music, video, bets, quizzes, etc.)
- Shopping services (shopping via mobile devices, price comparison, etc.)
- Medical and health services (medical monitoring for patients, mobile intranet for MDs, diet or nutritional tips, etc.)
- Educational services (seminars for commuters, edutainment for children, sightseeing explanations, etc.)
- Work-related services (mobile intranet, virtual secretary, logistics, parcel tracking, etc.)
- Multimedia communication services (video conferences, communities, etc.)
- Telematic and security services (fleet management, navigation, controlling and monitoring surveillance systems or intelligent houses, etc.)
- Public services (public information, registering for services like bulky waste, etc.)
- Customer services (remote diagnostics, air pressure in tires, online manuals, callbacks, etc.).

Only a few of these services have found their way into the automotive domain and not all of them seem to make sense in an automotive environment.

Back in 2001 consultants said that the killer application for Internet in cars by 2004 would be shopping and online banking and that "consumer demand will lead to some kind of e-vehicle wonderland in 2004, where the choices of services will be huge" [3]. As it turns out, most consumers are not very interested in Internet in cars today.[6]

The J.D. Power Report every year portrays customers' most wanted features in cars in a ranking. In 2003 the three reports for Europe, USA and Asia Pacific revealed the findings shown in Table 1. [12, 13, 11].

[5] According to a study from the Institute for Future Studies and Technology Assessment, IZT [10].

[6] Why should someone try to pay bills or shop in a car when they do not do it with their mobile phones, when they are not distracted by traffic and have both hands free?

Table 1. Ranking results of Automotive Emerging Technology Studies

Feature	Europe	USA	Asia Pacific
Rear-seat entertainment	11	23	12
Satellite Radio	9	21	–
Digital Broadcast Radio	–	12	–
Wireless Connectivity	8	16	11
Flexible Format Audio System	7	18	–

Customers are generally more interested in safety-related than in entertainment technologies. Night vision systems and external surround sensing are the most wanted features in cars today.

2.2 Pricing and Billing

In this section various issues of pricing and billing will be discussed. This will be more a practical than a theoretical or academic approach since there are thousands of academic books and theories dealing with economic and marketing issues.

If it comes to product or service development one of the first questions needs to be: How do we generate revenues with this? One of the major rules for selling services (especially online or virtual services) is *bill it or kill it!* If you cannot bill the service that you want to sell then it might be a better option not to sell it at all. These are the (very simplified) foundations of a business model for online or streaming services.

The growth rate for the acceptance of a new technology correlates *mainly* with two variables:

- Price and
- Function.

Price is the cost of the hardware as well as the fees or subscriptions for additional services. The function is dependent on added values for the customer, quality, benefits, new technologies, pervasiveness and the total availability of services.[7] The customers will not adopt a service where price and benefit (i.e. function) are imbalanced. Some products and services suffer from their complicated product descriptions or handling, poor benefits or bad quality, while others keep the customers away with high prices. Availability and pervasiveness are key factors too. A DVD player would be quite useless if there was one store or rental station every 300 km.[8]

[7] New technologies such as Bluetooth, General Packet Radio Service (GPRS), Wireless Local Area Network (WLAN) or Universal Mobile Telecommunications System (UMTS) will presumably generate new products and services.

[8] Video telephony is another good example: stand-alone devices have been on the market for many years yet there are very few people owning one. First it was

A good example of a "bad" service is the Wireless Application Protocol (WAP). WAP allowed the user to connect to the Internet via a cellular phone, but WAP is still not successful owing to complicated use, high costs and a lack of benefits.

The pricing model for an ICE service can range from monthly or annual subscription fees, pay per view, volume-based fees to time-based fees.

Pricing a service – especially an online service – can be a tough problem in a new market when there is no or not much knowledge about possible customers.[9] One method to find a price for a service is *target costing*, for example. Target costing is a method where the realizable price on the market for a good or service serves as a basis. The production costs (or costs for the preparation, provision and delivery of media) have to be optimized to meet the realizable price – it is a calculation of costs done backwards. The realizable price on the market can be determined through market research.

Telecommunication companies usually subsidize cellular phones when contracting a new customer. The companies hope that customers will generate enough ARPU (average revenue per user) to turn the subsidization more or less into a payment by installment. The same could happen in the ICE industry. Audio or video receivers can be sold below their wholesale price with a 2-year contractual binding period for content delivery with a particular subscription rate.

The billing should be easy to understand and transparent for the customers with payment options like bank collection, credit card or bank transfer. The billing system should include or be able to cooperate with a customer relationship management. Security is a very strong enabling technology for billing purposes because the non-repudiation for consumed services is most important for both customer and service provider.

Sirius Satellite Radio Inc. (a provider for satellite radio) is one of the few companies that are quite successful with their ICE business model. Sirius had more than 350,000 subscribers in the first quarter of 2004 and their *quarterly* growth rates range from 34% to 127%. Sirius provides more than 120 channels with music (primarily ordered by style), sports, news and entertainment without any commercials. The customer has several options for a subscription including a lifetime subscription with a single payment.

OnStar has a very successful personal assistant service via cellular phone installed in the car. The service can include driving directions, emergency services, online concierge or even stolen-vehicle tracking. OnStar offers a basic package for $16.95 and a premium package for $34.95 and has over 2.5 million subscribers.[10] If OnStar could lower the price for the basic package down to

---

expensive (not only the device but also the charges for each call) and second there was no one else around with such a phone. The Internet with its webcams and video telephony solutions showed that there is a demand for such features.

[9] Nobody predicted the success of the Short Message Service (SMS), for example.

[10] `www.onstar.com`

$10 per month nearly 50% of all consumers would be interested in this feature and at $6 per month nearly 60% would be interested [13]. This clearly shows the importance of good market (i.e. pricing) research.

Not everything in security can be backed up by a business model. Like with airbags or ABS the customer will anticipate at a certain point the existence of a specific feature or technology in his car. Secure updates for a car with a lot of IT systems should be possible but it is arguable whether customers are willing to pay extra money for such a feature. Nevertheless it seems necessary to integrate these abilities at least for debugging and patching purposes.

3 Standards and Technical Issues

One major problem for new technologies can be a missing standardization. Proprietary systems rarely have the ability to reach high market penetrations especially technologies that require a large infrastructure like mobile telephony.

There is a common problem when services are dependent on an infrastructure: without infrastructure there are no services and without services there is no need for an infrastructure. But since the build-up of an infrastructure from scratch is expensive, OEMs, suppliers and providers should agree on common frameworks that allow the interoperability of each solution to protect their investments.

Ideally the transmission for streaming media should work everywhere, even if the vehicle is driving through tunnels, deserts, low valleys, canyons or cities with tall buildings. There are countries with vast sparsely populated areas and crowded urban areas as well.

The typical maximum bandwidth requirements in bits per second (bps) are shown in Table 2 [4].

Table 2. Maximum bandwidth requirements

Service	Bandwidth
ITS Mobile Data, GSM, FM radio and IR beacons	10 kbps to 1 Mbps
MP3 (standard quality)	128 kbps
Digital CD Audio	1.4 Mbps
Digital DAB Audio	1.5 Mbps
Navigation Map Data DVD ROM	4.2 Mbps
Digital Video (MPEG2)	5-10 Mbps

These requirements have to be compared with the maximum bandwidth of available technologies. Table 3 shows a comparison of maximum bandwidth and maximum range for each transmission technology.[11]

Table 3. Maximum bandwidth and range

Technology	Max. Bandwidth	Max. Range
GSM	9,6 kbps	35 km
GPRS	107,2 kbps	35 km
UMTS	2 Mbps	35 km
Bluetooth	477 kbps	100 m
WLAN	108 Mbps	300 m
DVB-T	31.7 Mbps	65 km
Satellite	155 Mbps	several thousand km

The combination of those two tables leads to the conclusion that only UMTS, DVB-T (or DVB-H respectively) and satellite transmissions are capable of delivering real-time audio or video in an appealing quality (depending on the size of the screen).[12] Bluetooth and WLAN might be good for hotspots like gas stations, motels or restaurants, but not for moving vehicles by reason of their short range and limited capabilities.[13]

Several possibilities for content delivery like point-to-point, point-to-multipoint or broadcast are directly linked to the business model and the required infrastructure. The decision for a specific content delivering infrastructure has to be made very carefully, since an initial build-up of such an infrastructure can cost up to several billion dollars. It has to meet five criteria that were originally for cellular network providers, but are also valid for in-car entertainment infrastructures:

- Coverage (of area)
- Cost (planning, building and running costs)
- Capacity (bandwidth, availability)
- Capability (scalability, compatibility)

[11] The range is dependent on sending radio beacons, radio cell stations, etc. and not the receiving (mobile) device. There is a maximum bandwidth of 2 Mbps for UMTS but it is not sure yet if providers will support this, due to the fact that they announced a *usable maximum* of 384 kbps. All statements show the theoretical maximum bandwidth and maximum range (independent from another).

[12] For an automotive DVB-T solution see [17].

[13] Technically this is no real streaming because the user has to download the media completely at the hot spot. The consumer might still start watching it while downloading but he/she also might experience difficulties due to the fact that there is no dedication of bandwidth for streaming media in the associated 802.11x protocols of WLAN.

- Clarity (no drop-outs or artifacts).

Furthermore there are other technological issues. One of the most significant topics is the user interface. In many countries legislation prohibits to write or read emails, watch TV or to use a non-hands-free mobile phone while driving, hence there have to be other assisting technologies. Voice recognition and text-to-speech are the favored technologies for an interface solution.

Voice recognition needs high computing power and a lot of memory since it has to analyze not only words but grammar and syntax as well to understand complete sentences. This can be avoided by transferring the analysis to a server (via GPRS or UMTS for example) with higher computing power and specialized voice recognition soft- and/or hardware [3].

In a personal computer environment voice recognition works pretty well when the user takes the time to train the program to his voice and pronunciation. In an automotive environment complexity grows, because more challenges are added to voice recognition:

a. A car is a noisy place. Airstreams, sound of rolling motion (tires), engine, other cars or fellow passengers create a continuous background noise which has to be filtered out for voice recognition. This has to work even in a convertible or truck.
b. Different drivers. Voice recognition in a family car that is used by two or more drivers must be able i) to differ between all persons and ii) to "understand" each one of them, even if they have a different dialect.

Legislation will affect development speed of voice recognition interfaces and simplified cockpits.

For a mobile office or Internet in cars the communication system must be able to read text aloud. The user has to keep his hands on the wheel and his concentration focused on the road.

DVB-T and DVB-H have a very good chance of conquering the mobile infotainment market in Europe. By 2012 it should be widely available in Europe and the receivers will not be too expensive. Additionally the user does not need a satellite dish – a small antenna will do the job.

Satellites would be a good alternative for ICE but they have one important disadvantage: the need for a satellite dish tracking a geo-stationary satellite continuously. Other alternatives such as Bluetooth or WLAN would be quite expensive if infrastructure has to be build from scratch. Mobile technologies like the Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS) or Enhanced Data rates for GSM Evolution (EDGE) are too slow for video signals in a pleasing quality and/or yet too costly for the user.

4 Security Issues

This chapter will discuss a few security issues in ICE and other automotive business models. This will be done more briefly and in general because most technical problems and solutions are presented in separate sections of the book, like security in automotive bus systems in [6], security aspects of mobile communication protocols in [7] or DRM-related solutions in [5] for example.

Nowadays customers have to have security concerns about many technologies. Even modern cellular phones are at the risk of catching a virus. Some devices are capable of flashing their EEPROMs through CD, DVD, memory card or other wired or wireless connections. Computers are exposed to various threats once connected to the Internet. This does not only concern the car owner or service subscriber but the content provider as well.

One major security problem is the connection of car entertainment systems with a large public or open network like the Internet, the Public Switched Telephone Network or wireless technologies like WLAN or Bluetooth. If the security measures are not well implemented a hacker can manipulate, destroy or fake data.

Some of the services can turn against their users. Location-based services are capable of tracking every move of a person. Stalkers, kidnappers or thieves could find this information very useful.[14]

Even in a one-way communication, such as satellite TV, radio or other broadcasts, attackers can do a lot of damage. This is most important in the area of pay per content. If an attacker should succeed in deceiving the billing mechanisms of a content provider the attacker might get the content for less than the actual price or even for free.[15] Even worse, the content could appear on the bill of other "innocent" service subscribers. The damage to the image of the company can become huge in such a case.

A worst-case scenario is the invasion of the car by an attacker. The (electronic) complexity of cars is constantly rising. A car like Volkswagen's top model, the Phaeton, has more than 60 microcontrollers networked over three busses, subbusses and one optical bus with a total length of over 3.8 km of cable and more than 50 megabytes of memory [8]. Cars are nowadays complex systems with more combined computing power than a standard personal computer. It takes a lot of IT and engineering knowledge to understand and control the interactions between the systems. If an attacker finds a way to "smuggle" commands via wireless technologies from one device (a navigation system for instance) to another (like the motor control unit or the braking controllers, maybe via CAN) this can cause serious harm to passengers or innocent bystanders by manipulating the brakes and related assisting systems or damaging the motor by changing the characteristic curves. Every car manufacturer knows about this risk but the research and development in this field

[14] The tabloids might pay a lot of money for this kind of information on celebrities.
[15] This a common problem for Pay TV providers: attackers manage to get the service for free by breaking or bypassing the encryption scheme.

is in its infancy since this is neither their core competency nor the supplier's. The good thing though is that these security requirements can lead to a higher security level of many other devices due to spreading awareness, knowledge and the availability of corresponding microcontrollers with built-in security features.[16]

Manipulated media could flash the EPROM of a device, turning the automotive computer system into an open relay for spam mails which get sent via GSM, GPRS or UMTS at high cost. The same can happen when the car PC receives emails contaminated with viruses, worms or trojan horses. It could also be an attacker via WLAN, Bluetooth or a cellular phone which infiltrates the system with a specific code that gives him the power to control the car PC or a specific subfunction.

A very common mistake is the use of proprietary security measures such as self-developed cryptographic algorithms. Only "public" algorithms (in the meaning of "published" or "freely available") that have been evaluated by the cryptographic community should be used. This is equally true for Digital Rights Management (DRM) systems. Poor protection schemes often led to security breaches in the past. The consequences of such an incident can be devastating. Imagine a company that gave away millions of DRM-protected files like movies or songs and after some time somebody is breaking the inherent protection scheme. The losses can become huge not only because nearly all media that has been given away has to be considered "lost"[17] but also for re-encrypting all the existing media, not to mention the irrecoverable trust of investors, partners and consumers.

Privacy is an often neglected feature in security design or implementations. Especially German customers can become very sensitive about personal information. If the process requires the recording of personal data at least pseudonymity should be granted as well as the assurance that the data will not be sold or given away to third parties or be used for marketing purposes unless permitted by the customer.

Some non-ICE services give a good understanding of how one should *not* handle security issues. A service like OnStar's "Remote Door Unlock" might come in handy in cases, but it is also an easy-to-use tool for car thieves. The only thing the attacker needs to know is the PIN (a 4-digit number) to get the door unlocked remotely by an operator from the OnStar service. OnStar offers also a service for "Stolen Vehicle Tracking" where the customer can call the service, provide his PIN code and let OnStar locate his car. This might be useless if the attacker has enough time to disable the cell phone connection. Since the PIN can be changed with a simple phone call (name old PIN and give

[16] The car manufacturing industry should not repeat the same mistake as the computer industry. Security is not a simple add-on it has to be a fundamental design principle.

[17] If the business model involved pay per view or pay per listen consumers can circumvent the protection and view or listen to the content without charge and/or distribute the content to third parties.

new PIN) the attacker could alter the PIN and the car owner would not be able to report the theft without knowing the new PIN. This is a good example of bad security. It seems that hackers even found a method to circumvent the GPS receiver in OnStar's device thus customers do not have to pay for the service and can use it with their car PC. This indicates how important a good holistic security design is.

An old saying is: the chain of security is only as strong as its weakest link. From this it follows that not only the encryption scheme is relevant but all other involved components and processes, like IT systems (middleware, front- and back-ends), devices, busses and cables, diagnostic connectors, secure Internet connections, defined services before customer, after sales, etc. The bottom line is: security is a *process* not a *product*! Therefore the complete value chain has to be secured to render attacks impossible.

Unfortunately there is no universally valid solution to security challenges because it depends on the particular business case, hardware, process chain, etc. Every security measure has to be verified on a regular basis. Protocols or algorithms that once were considered secure might get compromised due to new sophisticated attack methods[18] or the advances of computing power in freely available hardware. Security solutions for one ECU might not apply as well in another ECU.

5 Summary and Perspective

The revolution in ICE is already here, but only a few motorists seem to care. The business world is still excited about the profit potential and the technological benefits. The market for ICE will become bigger and enriched by new devices and features. But it is still unclear how fast customers in Europe will adopt this because of many uncertainties.

Even if a service like streaming media can be provided everywhere via satellite, radio broadcast or any other technology, it is still questionable whether people will adopt the service when they have to pay for it. Video rental via Internet for instance has not fulfilled the expectations. The Traffic Information Service (TMC), which is transmitted via FM in the Radio Data Service (RDS), has been quite successful in Europe due to the fact that it is available free of charge. The willingness to pay for telematic services in Europe is very low, even for services that are considered as useful by motorists [16] [3]. Only the offboard navigation is regarded differently because the fees are comparable to the costs of a new CD for onboard navigation. This consumer

[18] For instance, the new attack methods against hardware with cryptographic algorithms since the mid 90s like the measurement of energy consumption or electromagnetic radiation to extract the secret keys from the hardware.

behavior does not apply to the USA where service subscriptions are generally more accepted.[19]

This is equally true for Internet in cars which is not a serious technical problem (except for high bandwidths). Europeans are not interested in online services in cars yet. The industry is still searching for the "holy grail": a killer application, that would make Internet in cars indispensable [3].[20] Customers are not willing to pay the high costs for the hardware and the intermittent wireless connectivity. Additionally there are safety concerns since the industry has not fulfilled the expectations regarding the user interface (voice recognition and text-to-speech).

The legislation in nearly any country worldwide prohibits the driver watching TV or video while driving [15]. This leads to the conclusion that (video) ICE is not suited for customers that drive most of the time alone or do not have a family.

As a matter of fact ICE today is nearly synonymous with the *product* range "rear-seat entertainment". The only *services* that have the potential to become profitable soon are satellite radio and personal assistance services like OnStar. If DVB-T is ready for ICE other companies offering pay per view or TV via subscription will have a very strong rival since DVB-T is free of charge.

Finally the industry has to care for a safe and secure car as a comprehensive system. It will take some time and maybe the combined efforts of the car manufacturing industry to create such an environment as a base of operations for ICE products and services. It is necessary that the main focus is on the benefit of the customer and not on the business case.

The next 5 years will show if things like data stations and hotspots (at points of interest, gas stations, restaurants, motels, etc.) or streaming media services will emerge but there is the common problem: which came first – the chicken or the egg, i.e. without infrastructure there are no services and without services there is no need for an infrastructure.

But even if ICE is still suffering from unsolved technological problems and a lack of acceptance, automotive services are a very hot topic right now. OEMs and suppliers will and should continue their quest for new market potentials but they should be aware of the inherent risks that can arise from a lack of security.

[19] People in the US spend much more time in cars (approx. 90 minutes a day), go on longer travels and have larger vehicles like SUVs or minivans [14].

[20] The same happened with UMTS a few years ago – the killer application has not been found there either.

References

1. Eberhard Abele, Philipp Radtke, and Andreas E. Zielke. *Die smarte Revolution in der Automobilindustrie*. Redline Wirtschaft bei ueberreuter, Frankfurt/Wien, 2004.
2. Bernd Beckert, Andre Jungmittag, and Paul J.J. Welfens. Internetwirtschaft 2010 – Perspektiven und Auswirkungen, 2003.
3. Elizabeth A. Bretz. The Car: Just a Web Browser with Tires. *IEEE Spectrum*, pages 92–94, Jan. 2001.
4. C. Ciocan. The Automotive Audio/Video – Telematics Data Bus Technology – Politics – Standard. *VDI Berichte*, (1547):717–720, 2000.
5. Marko Wolf, André Weimerskirch, Christof Paar. Automotive Digital Rights Management Systems. This book.
6. Marko Wolf, André Weimerskirch, Christof Paar. Security in Automotive Bus Systems. This book.
7. Jan Pelzl, Thomas Wollinger. Security Aspects of Mobile Communication Protocols. This book.
8. J. Fehrling, A. Heinrich, K. Müller, A. Paggel, and I. Schneider. Versionsmanagement für Transparenz und Prozesssicherheit in der Steuergeräte-Entwicklung. In *VDI-Berichte Nr. 1789*, VDI-Berichte. VDI-Verlag, Baden-Baden, 2003. S. 219–230.
9. Gesellschaft für Unterhaltung- und Kommunikationselektronik. *Vom ersten Autoradio zum mobilen Multimedia-Center*, June 2004. `www.gfu.de/pages/history/his_auto.html` am 04.06.04.
10. Institut für Zukunftsstudien und Technologiebewertung (IZT), Berlin. *Entwicklung und zukünftige Bedeutung mobiler Multimediadienste*, 2001.
11. J.D. Power and Associates. *Consumers in Japan Express Strong Interest in Hybrid Electric Vehicles*, Sep. 2003. `www.jdpa.com`.
12. J.D. Power and Associates. *European Consumers Express High Interest in New Automotive Technologies, But Interest Drops Significantly Once Price is Introduced*, Sep. 2003. `www.jdpa.com`.
13. J.D. Power and Associates. *Several Emerging Automotive Features Garner Strong Interest from Consumers at Expected Market Prices*, December 2003. `www.jdpa.com`.
14. Car Video Takes Off in the U.S. *The Hansen Report on Automotive Electronics*, 14(1), Feb. 2001.
15. Das vernetzte Auto. *ElektronikPraxis*, (15):24–27, August 2001.
16. Wolf-Henning Scheider. Digitale Empfängertechnik für Telematik und Unterhaltung. In *Technischer Kongress 2003*. Verband der Automobilindustrie, 2003.
17. Axel Zimmermann. Altera helps Hirschmann with mobile TV. *Automotive Engineering International*, (10):84–86, October. 2004.

In-Vehicle M-Commerce: Business Models for Navigation Systems and Location-based Services

An Analysis of the Economic and IT Security Implications

Klaus Rüdiger and Martin Gersch

Institute for E-Business Security/Competence Center E-Commerce
Ruhr University of Bochum, Germany
{klaus.ruediger, martin.gersch}@rub.de

Summary. The introduction of new, innovative business models for in-vehicle m-commerce requires the application of advanced IT security measures and has strong economic implications. In this article, the authors analyze the most important IT security and economic implications and use the practical example of an innovative business system for navigation systems and location-based services. This in-vehicle m-commerce business system has been introduced by one of the leading suppliers of aftermarket navigation systems. The analysis shows that when innovative in-vehicle services are introduced, the revenue generation may shift from hardware devices to service revenues and new competitors are becoming relevant. They offer, for example, user-centric services with the help of mobile devices. Basic requirements of sketched developments are applications of advanced IT security measures such as Digital Rights Management systems.

Keywords: in-vehicle m-commerce, navigation systems, business models, IT security, digital rights management, location-based services, electronic business

1 Introduction

Mobility is considered a basic need in modern society. Since the invention of the car, the strong desire for mobility is reflected in the continuously growing number of vehicles and kilometers driven.[1] With the introduction and widespread use of cellular phones, mobility nowadays also includes the ability to use selected services at any time and at any place ("m-commerce"). For

[1] In Germany the number of passenger cars increased by more than 64 % from 1982 to 2002, and the number of driven kilometers has grown in the same period by over 66 % and this year reached 528 billion kilometers [3, 22].

many people, it seemed obvious that these two aspects of mobility were compatible and therefore it was only a matter of time before they were brought together. Consequently, the m-commerce hype was followed by the "in-vehicle m-commerce" hype at the end of the last century. The potential demand for in-vehicle services seemed to be endless: shopping, banking, Internet, dynamic navigation, automatic airbag notification, vehicle tracking, and brakes by GPS are all examples [28]. But only a few years later came disillusion. There was little or no demand for most of the services offered [15, 20, 28, 33]. Walter Maisel from Siemens VDO explained: "We were saying everything is possible ... browsing through the Internet, the latest surprise from the stock market. But that's not the highest demand for a car driver. We mixed too many things together... Telematics and navigation systems must offer services that enhance safety and don't charge for what is available free on the radio or the internet." [54].

Despite the slow development of in-vehicle services, VDO Dayton, a brand of Siemens VDO, launched a very innovative business system for in-vehicle services in 2002 called C-IQ – Intelligent Content on Demand. This consists of vehicle navigation and related location-based services. For the first time, the customer is offered a large product range consisting not only of dynamic navigation for all European countries, but also a large number of travel products (travel guides, hotel and restaurant guides, shopping guides, etc.). The customer can choose the desired content and the activation period and only has to pay for the time he uses the selected service. For a holiday in France, for instance, the customer can order the French road map for a one-month period for 19.99 euros. This innovative "pay per use" system (conditional access) could be realized because of the employment of advanced IT security measures. The data on the CD-/DVD-Roms is encrypted and the functionalities of a Digital Rights Management system are employed to control access to the services.

The focus of this article is an analysis of the C-IQ business system. In addition, the economic implications the IT security requirements which arise through the change of the business system are analyzed. The importance of IT security is growing with the introduction of new m-commerce business systems and further in-vehicle services [30, 33]. IT security can be seen as a prerequisite not only for the introduction of the C-IQ system, but for nearly all innovative telematics applications. Here, a detailed insight into the C-IQ business system is given and the implications, opportunities and threats are analyzed.

2 Business Models and Business Systems

The description and analysis of business activities become quite comprehensive and complicated when based on innovative and complex approaches, which comprise a vast number of underlying determinants and business ac-

tors. Many examples of such innovative and complex approaches can be found in the e-business sector [56]. Business models are primarily used as a useful tool to describe the core elements of such complex business activities in a simple and transparent way and therefore offer an insight into how business is done, particularly how sufficient revenues are generated [37, 56]. According to Timmers and Wirtz a business model can be defined as a description of a company's production and service system. It describes in a simplified and aggregated form the flow of resources, the way they are transformed in the production process and the information, products and services a company offers to generate revenue. A business model indicates the most important aspects of a company's strategy and the roles the involved actors assume [52, 56].[2]

To reduce complexity each business model can be divided into several partial models [56, 23].[3] Depending on the business activities to be considered the analysis can focus on the most relevant partial models.

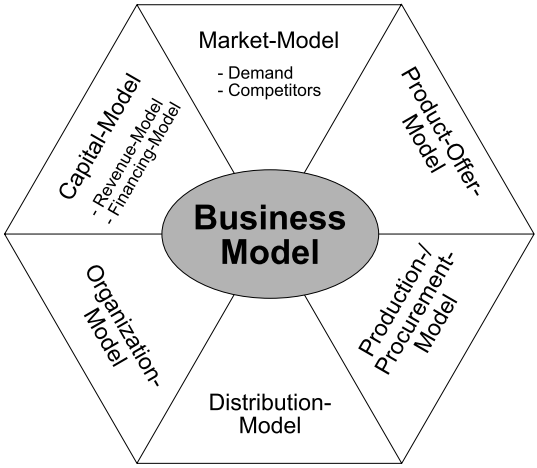

Fig. 1. Business model [56, 23]

With the help of the market model the relevant markets for the business model can be defined. For each relevant market the current and potential

[2] Neither in the German nor in the English literature is there a common understanding and definition of the term "business model" [36, 37, 52]. Both Timmers and Wirtz offer well explained definitions. The definition used is mainly based on Wirtz, since it includes as far as necessary also strategic aspects and offers partial models for a more systematic and in-depth analysis. According to Timmers a business model is part of the marketing model which additionally includes the marketing strategy of the business actor under consideration [53]. See also [37].

[3] The partial models of Wirtz were modified and completed by the organization model. See [23]

customers, their demand for products and services, and their willingness to pay are identified as well as the direct and indirect competitors. The product offer model describes which combinations of products and services are offered to the different customer groups. The production and procurement model analyses the procurement of input from different suppliers and all aspects of the transformation of the input into sellable products and services for the market. The entire process of distribution – comprised of the different channels of distribution, the distribution agreements and networks – is analyzed in the distribution model. Organizational aspects are considered in the organization model. The capital model consists of the revenue and the financing model. The financing model describes from where the financial resources are derived to cover the financial needs for the business activities. With the help of the revenue model the sources and forms of revenue generation are analyzed. Many business models fail because the revenue does not grow as fast as expected.

For the e-business sector various typologies of business models have been developed, but each applies different criteria [37]. Using the products and services offered as classification criteria (product offer model) business models for in-vehicle services could be classified into: (1) vehicle related basic telematics services such as automatic emergency and collision notification, roadside assistance, vehicle tracking or remote diagnostics; (2) navigation systems and related location-based services; and (3) infotainment, communication and transaction [20, 56]. The different types of business models can be implemented in a variety of ways [36, 52]. Therefore a specific implementation of a general business model is called a business system [24, 1]. VDO Dayton's navigation system "C-IQ" represents a specific business system of the general business model "navigation systems and related location-based services". As the following analysis will show, VDO Dayton has chosen an innovative way to implement a business system for navigation systems and location-based services – quite different from the business systems of its competitors. In the next chapter, the most important aspects of each partial model of the VDO Dayton business system are analyzed.

3 The C-IQ Business System

3.1 Market Model

The relevant markets can be identified by use of the theory of the gap in the chain of substitution, which means basically that all available products which consumers consider as a possible substitute for the C-IQ system belong to the relevant market [38, 39]. The C-IQ system offers navigation, travel products like electronic restaurants or city guides in a web-based format with numerous photos and detailed descriptions plus the ability to locate points of interest. Recently, additional products have been added, for example a camping and caravaning guide, a shopping guide, a business directory and speed camera

info.[4] The integration of personalized services, infotainment and m-commerce (transaction) is planned for the future.

Competitors

In Europe presently the core application for the driver is still navigation with dynamic rerouting [10, 25, 28, 54].[5]. All important suppliers of aftermarket vehicle navigation systems and navigation radios offer this feature in a reasonable quality and can therefore be considered as direct competitors for VDO Dayton.[6] For the German market the major competitors are Alpine, Becker, Blaupunkt, Pioneer and Panasonic [5]. Aftermarket vehicle navigation systems offer the advantage of large color monitors (widescreen monitors) which can be positioned on the dashboard in the sightline of the driver. This makes navigation comfortable and enhances safety. Connection of a wide range of further devices like a TV tuner, Internet browser, DVD player, additional monitors for the passengers or a handsfree car kit for the cellular phone is possible [14].

Navigation radios have only a small display, but offer a radio and CD player.[7] Additionally top navigation radios provide a handsfree set with SIM-card, internet and e-mail access. Here, too, the connection of further devices is possible. The radio's main advantage is that it is easy to plug in to the dashboard in the 1 DIN slot [2].

Another group of direct competitors are the factory-installed navigation systems offered by the car makers. These systems are integrated in to the dashboard and are usually connected to further electronic devices like on board computer, air conditioner or audio system and can be operated with the controls on the steering wheel. Usually they use the display of the on board computer and, if available, the head-up display. Their major advantage is the possibility to offer basic telematics services, like automatic emergency call, vehicle tracking or remote diagnostics [28]. Nearly all factory-installed navigation systems used by OEMs such as Audi, Volkswagen, BMW or Volvo are derived from the same suppliers as the aftermarket vehicle navigation systems [25, 59]. Siemens VDO supplies navigation systems to, among others, the Volkswagen Group and BMW. Further suppliers are Harmann International

[4] These products have been available since summer 2004.

[5] Dynamic rerouting uses real-time traffic related information from the Traffic Message Channel (TMC) to identify problems and calculate an alternative route on request, thus avoiding certain hindrances, e.g. traffic congestion. The TMC service is available in virtually all West European countries. For a more detailed description see [13] 39-40; [19]

[6] Products which do not offer exact positioning combined with navigation to the desired location can be excluded from the relevant market since they do not offer an equivalent use as navigation systems [32].

[7] Some 1 DIN navigation radios offer large splitscreens (slide-out monitors) [1].

and Robert Bosch GmbH, who provide the systems for DaimlerChrysler and Mercedes-Benz [59].

Due to the fast technological progress, new competitors have recently entered the market for navigation systems offering solutions which do not need to be connected to the vehicle: personal navigation devices, PDAs (personal digital assistants) and cellular phones. Just two years ago, it seemed impossible that these devices could offer navigation services at a reasonable level [4, 20, 49].

Personal navigation systems cannot compete with the leading aftermarket vehicle navigation systems but they offer navigation with dynamic rerouting at a sufficient quality level and a competitive price [4].

For PDAs, two different solutions are available. The "onboard systems" are equipped with GPS and digital maps stored in a chip card. Routes are displayed on the screen and audibly explained via speakerphone. The major disadvantage is that dynamic rerouting is not available. Additionally the "offboard systems" require a cellular phone. The desired destination is transferred via cellular phone to the service provider, who calculates the route and transfers it again via cellular phone to the PDA. No digital map database is required on the PDA, which means that updates are not necessary. The system always refers to the current digital map databases of the service provider. The first navigation-cellular phone for the German market, the Wayfinder 2.0, also operates as an offboard system. All offboard systems offer dynamic rerouting via cellular phone. Like personal navigation systems, both mobile onboard and offboard systems cannot compete with leading aftermarket vehicle navigation systems but do offer navigation services at a sufficient quality level [4, 5]. The possibility to connect further devices is very limited compared with fixed in-vehicle navigation systems.

On the other hand, mobile navigation systems offer some substantial advantages compared with fixed in-vehicle solutions. As they are user-centric and not vehicle-centric, customers can take the service with them from one vehicle to another (e.g. for families with several cars, business men using a company or rental car without navigation system) or use them for trips made by different modes of transportation, which is called multimodal routing [28, 48]. Combined with location-based services such as a restaurant finder or sightseeing tours with information about objects of interest, mobile navigation systems are even a useful tool for pedestrians [18, 32]. In fact, with the exception of basic car-related telematic functions, most services like navigation, location-based services, further personalized services or infotainment can also be made available for use outside the vehicle; although many applications do require a certain minimum screen size, not only for the use of the service but also for control and safety reasons [20, 28].

This short overview shows that through technical progress the relevant market and the number of competitors are growing. In the future the systems will offer more functions and services for the driver and passengers combining navigation, location-based services, personalized services and infotainment.

VDO Dayton has already taken an important step in this direction by offering new services (electronic hotel, restaurant and city guides etc.), and introducing a new pricing and billing system (per use charges) which allows the identification of the customers and thereby the introduction of services according to the individual customer's needs. As a result, the border of the relevant market will fade away and the number of indirect competitors offering parts of the new services will grow further.

Demand

The demand for navigation systems varies strongly among the different regions. About five to 10 % of new car buyers in Europe order models with onboard navigation, compared with 15 to 20 % in Japan and two % in the USA [18, 21, 59]. Two major reasons for the low sales figures in the USA are the higher prices compared to Europe and Japan and the later introduction of dynamic rerouting into the market [18, 25, 54]. The major reason for the relatively high sales in Japan is the country's lack of a coherent address system. Toyota Motor Corporation estimates that half of its cars in Japan are equipped with navigation systems [54].

Until now, both aftermarket suppliers and car makers considered their major potential customers to be owners of luxury vehicles, which was reflected in the pricing strategy and the positioning [58, 21]. Therefore, in the eyes of many customers, the prices for navigation systems were too high. Recently, a strategy shift can be observed. Increasingly, the owners of other types of vehicles such as upper middle class cars or sports utility vehicles are considered to be additional potential customers, as well as users with a "technical affinity" regardless of the type of vehicle they drive.

3.2 Product Offer Model

Since the introduction of the C-IQ system in the second half of the year 2002 the offered products and services have grown considerably. For a systematic overview the product offer can be divided into two interdependent parts:

a. The vehicle based system platform, which consists of all hardware components in the vehicle and the operating software; and
b. The (coded) C-IQ based services, which can be used by registering in the C-IQ business system (application software, database).

At present VDO Dayton offers five different aftermarket system platforms for the use of the C-IQ system. Three of them are stand-alone systems consisting of a DVD-Rom drive unit, including processor and controller, and a separate monitor. For dynamic rerouting an additional TMC receiver is required. The other two systems are 1 DIN navigation radios with a CD-Rom drive. The TMC receiver is already included. For exact navigation, the hardware must be connected to the vehicle's speedometer. In order to use the

C-IQ based services the respective operating software must be installed and the DVD or CD must contain all necessary data (digital maps, guides etc.). All systems can be connected with a wide range of further devices (see Section 3.1), but these applications are not yet integrated into the C-IQ system.

The C-IQ based services are comprised of road maps and traffic info via TMC, a large variety of travel products, special products and travel packages, in which certain services are bundled [6]. In the German market the product assortment shown in Table 1 is available.

Based on customer segmentation, VDO Dayton offers travel packages for business men, lifestyle- and leisure-orientated persons. Each package consists of four travel products. Additionally, several product bundles are offered combining road maps of two or three bordering countries with a maximum of two further travel products. All the products can also be purchased separately. The product assortment for the German market is the most comprehensive. In other countries certain products are not available, but for all European countries (with the exception of the Czech Republic/Slovakia/Poland) the three travel packages are offered.

The customer can choose between roadmaps from the two leading service providers, the US company Navigation Technologies Corporation (NavTech) and the Dutch company Tele Atlas N.V. The time it takes to calculate the optimal route is an important quality factor. Each service provider has developed its own routing algorithm for precise navigation in a timely fashion [18]. The VDO Dayton systems belong to the fastest and most exact in the market [4, 50, 45]. All in all, VDO Dayton offers a unique product assortment with cutting-edge navigation. Until now, none of the competitors have offered similar travel services. An enlargement of the product assortment through cooperation with further content providers and the offer of personalized or vehicle-brand specific services is planned for the future.

3.3 Production and Procurement Model

To analyze the production and procurement model it is once again helpful to differentiate between (1) the vehicle-based system platform with its hardware components in the vehicle and the operating software and (2) the C-IQ-based services like digital maps or guides.

The final assembly of the hardware (vehicle-based system platform) is done in the production facilities of Siemens VDO. The parts and components are produced according to order by third suppliers (e.g. GPS-receiver, standard parts of the installation kits for the vehicles) or they are purchased on the market (e.g. screens or circuit boards). The development and production of the software is done by Siemens VDO. The CD- and DVD-roms are produced by a third supplier.

[8] In Europe the C-IQ Speed Camera Info-Service is only available in: Austria – Switzerland, Belgium – Holland – Luxembourg, Germany and the United Kingdom – Ireland.

Table 1. C-IQ product assortment in the German market [6]

C-IQ-based services	**Description**
Maps and traffic info	Austria, Switzerland, Belgium – Holland – Luxembourg, France, Germany, Italy, Denmark – Sweden – Norway – Finland, United Kingdom – Ireland, Spain – Portugal, Czech Republic – Slovakia – Poland, Europe, USA, Australia
Individual products	
Varta Guide	Information on hotels and restaurants (with 3500 photos) together with key information on towns and places
Michelin Guide	Information on hotels and restaurants based on the Michelin Red Guide and tourist information taken from the Michelin Green Guide
Merian scout premium	Information on hotels, restaurants, points of interest, golf, shopping tips, city guides, theaters and opera
Merian scout Sport & Leisure	Information on leisure facilities and leisure destinations arranged by category, shopping tips
Merian scout Arts & Culture	Travel guide with information on museums, places of worship, castles, palaces and many other attractions
ADAC Camping-Caravaning Guide	Information on around 5400 key campsites across Europe
C-IQ Shopping Guide	Shopping addresses including large department stores, supermarkets and small boutiques, banks, gas stations etc.
C-IQ Fast Food Finder	Information on the nearest fast food restaurants
Dun & Bradstreet Business Directory	Provides business information including e.g. credit-checked addresses (containing over 7 million business addresses in Europe)
C-IQ Prepaid	Road maps or travel guides can be purchased in advance. C-IQ Prepaid can be acquired in blocks of 5 or 20 days
Special products [8]	
C-IQ Speed Camera Info	Information about current speed limits and accident black spots (where e.g. fixed speed cameras are sited) along the planned route
Travel packages	
Business package	C-IQ Shopping Guide, C-IQ Fast Food Finder, Dun & Bradstreet Business Directory, Michelin Guide
Lifestyle package	C-IQ Shopping Guide, The Varta Guide, Merian scout Premium, Merian scout Arts & Culture
Leisure package	ADAC Camping-Caravaning Guide, The Varta Guide, Merian scout Sport & Leisure, C-IQ Fast Food Finder

All the C-IQ-based services offered are based on third party data, which stem from different content providers. The core application (the digital maps) come from Navigation Technologies and Tele Atlas, the two leading companies for the provision of digital maps.[9] Nearly all important competitors of both car makers and aftermarket suppliers use maps from these companies [18, 50]. Additional content is mainly procured from well-known suppliers such as Michelin, Varta, Dun & Bradstreet and the German Automobile Club (ADAC e.V.).[10] The content is not exclusively produced for Siemens VDO. Michelin, for instance, also sells road maps combined with its guides for the navigation systems of competitors and for use with PDAs and Internet-capable mobile phones [7]. Siemens VDO encrypts the data from the different content providers and integrates it on the CD- and DVD-Roms.

A specific software program is employed for the code generation and the creation of a customer database, where all customer data (e.g. age, sex, occupation, vehicle) and the user history of each customer is stored and the access to the services is administered. A website for customer information, registration and the product orders was created.

For the distribution of the products and services, after-sales services, C-IQ customer registration and further customer support, Siemens VDO has contractual relationships with different types of dealers (car dealers, specialized dealers, electronics stores etc.) and call center operators (who act as a hotline). Additionally, contractual relationships with mobile phone network operators exist for the provision of access codes via SMS.

From the Global Position System (GPS) information is needed for the navigator's controller of the navigation system. The GPS system consists of 24 satellites run by the American Department of Defense. It can be used free of charge by private persons or companies, with certain restrictions [27]. The real-time traffic-related information (TMC messages) for dynamic rerouting is derived from publicly or privately owned traffic information centers and other sources, such as highway authorities. Their information is coded by a service provider and transmitted digitally in the sideband of FM transmissions [13]. Use of this system is free of charge in most European countries.

3.4 Distribution Model

With the introduction of the C-IQ system the distribution model has become more complex and comprehensive. In general, navigation systems and related location-based services are not as easy to sell or use as a simple radio or a sunroof, for instance [12]. The introduction of the C-IQ system with its large product assortment and a new pricing system makes understanding of the

[9] For a brief overview of the production of road maps see [18].

[10] The Varta and Michelin hotel and restaurant guide-books are among the market leaders in Europe. The German Automobile Club is the largest in Europe with over 14 million members.

navigation system even more difficult and requires an explanation from the dealer.

Because of the necessary support and installation service the distribution of the hardware devices is done mainly by authorized dealers. Siemens VDO has agreements with car dealers of all the major brands, specialized dealers, chains specializing in vehicle accessories such as Auto Teile Unger or D&W and discount-chains selling all types of electronic consumer goods as Media Markt or Saturn.[11] If the customer orders a new car with a VDO Dayton navigation system it is fitted directly by the car maker.

In order to use the C-IQ-based services three steps are necessary: (1) the registration as a C-IQ customer, (2) the order of content and (3) the transmission of codes, which must be entered in order to use the selected content. Steps one and two can be done through an authorized dealer, via Internet or via the service center. The PIN codes (step 3) are transmitted by telephone, e-mail and SMS to the customer.

The DVD- and CD-Roms are automatically delivered free of charge to the customer's doorstep by the postal service (The update service only applies to contracts with a limited activation period, see Section 3.6).

VDO Dayton follows a multi-channel strategy in a "clicks and mortar" combination. For the hardware distribution and customer registration the emphasis lies in the dealer network of the car makers and the specialized dealers, since explanation and support – and in many cases installation – is required, which can be best guaranteed by a trained staff. For the order of content and the transmission of codes the emphasis shifts to "electronic" distribution and to the service center.

In Germany a further distribution channel exists via the German retailer Aldi.[12] The C-IQ-based services are offered in combination with the C-IQ-enabled navigation hardware from Aldi's brands, like "Medion". However, only contracts with an unlimited activation period are presently available.[13]

In the future, agreements with further cooperating partners for the distribution of travel products and prepaid cards are planned. Travel agencies could offer respective maps and travel products such as restaurant or shopping guides in combination with a hotel reservation; car rental stations could offer special rates including maps and travel products; or supermarkets or gas stations could sell prepaid cards. In many cases the integration of new products and services will lead to new distribution channels, as the channels of the new cooperation partners could be used as well.

---

[11] Auto Teile Unger (A.T.U. Handels GmbH & Co. KG), `www.atu.de`; D&W (D&W Auto, Sport, Zubehör Handelsgesellschaft mbH & Co. KG), `www.dundw.de`; Media Markt, `www.mediamarkt.de`; Saturn, `www.saturn.de`.

[12] With over 5000 stores around the world, Aldi is an international leader in grocery retailing, `www.aldi.de`.

[13] This information is derived from the German service center (telephone call with a call center operator on 29 July 2004).

3.5 Organization Model

The Siemens VDO Automotive Corporation was created in April 2001 by a merger of Siemens Automotive AG and Mannesmann VDO [11]. The company is a global supplier to the automotive industry with worldwide presence in all automotive markets. It designs, manufactures and sells integrated electrical components. 43,000 employees worldwide work at 130 different locations spread over 34 countries on five continents [41, 43]. With sales and orders of 8375 billion euros the company generated a profit of 418 million euros for the year 2003. [40]. The company is divided into four business units (Figure 2).

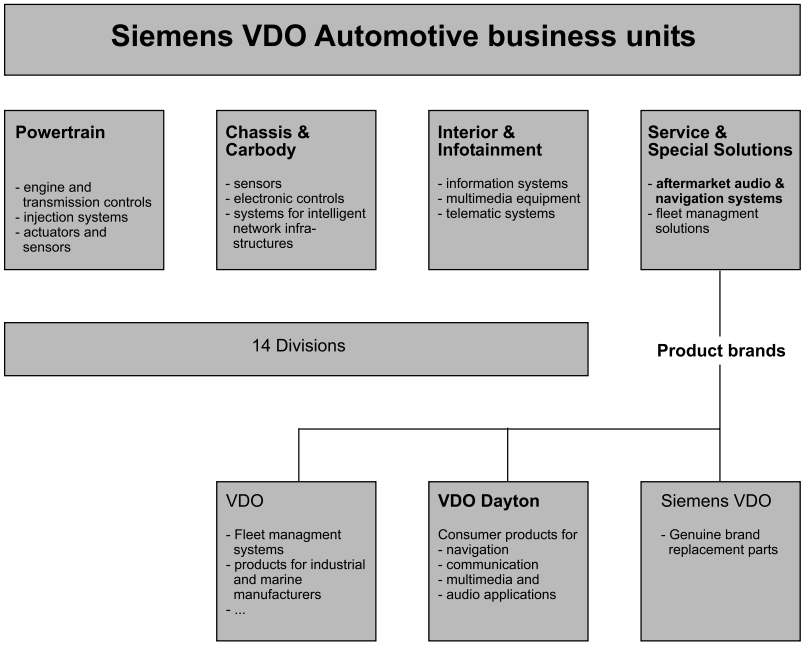

Fig. 2. Siemens VDO automotive business Units [42]

The first three business units (Powertrain, Chassis & Carbody and Interior & Infotainment) work directly with all important car makers. The Service & Special Solutions organization is responsible for trading with the automotive aftermarket and distributors. The products and services are offered under three different product brands: VDO, VDO Dayton, and Siemens VDO. VDO Dayton focuses on consumer products for navigation, communication, multimedia, and audio applications [42].

Navigation systems are only a small part of the wide range of products and services Siemens VDO offers to vehicle manufacturers and the aftermarket.

The overview of the organizational structures shows that the company has the competencies – and already provides certain solutions for all three classes of possible product-offers – for in-vehicle services: (1) vehicle-related basic telematics services, (2) navigation systems and related location-based services and (3) infotainment, communication and transaction.[14] The product range, the competencies, the worldwide presence and, last but not least, the fact that Siemens VDO is among the market leaders in navigation systems with a strong relationship to nearly all leading car makers, provided the basis needed to introduce the new business system C-IQ [11].

3.6 Capital Model

For the analysis of the C-IQ business system only the Revenue Model will be considered, as the introduction of new products and services combined with a new pricing system has strong implications for the generation of revenue.

Traditionally, the customer purchased a navigation system which included the hardware and a CD- or DVD-Rom with a current road map of Germany and bordering countries or even a complete map of Europe [4]. In addition to the hardware costs, which vary from nearly 1000 euros for the cheapest VDO Dayton DIN navigation radio to over 2000 euros for an aftermarket navigation system with a large color monitor, the customer had only additional costs during the use of the system, which was when he decided to buy a new CD- or DVD-Rom with updated map data.[15] Of course, there are also additional costs whenever the customer wanted additional products such as maps for other countries or travel guides. The customer knew the price he had to pay in advance regardless of how often he used his navigation system. This implied that the average price per use of the navigation service was determined by the customer and varied with the frequency of use. VDO Dayton generated its revenue mainly by selling the hardware and, to a small extent, by the replacement of CD- or DVD-Roms.[16]

With the introduction of the C-IQ business system, the customer is confronted with a more complex yet also more flexible pricing system, which perhaps better fulfills his individual needs. When the customer buys the C-IQ-enabled navigation hardware he can choose between two different types of contracts: [17]

[14] See Section 2

[15] The data of the digital road maps only reflect the local situation at a particular point in time. Since these environmental data change dynamically, 15 – 20 % of the data will change to some extent within one year [46].

[16] For many factory-fitted and aftermarket systems it is not necessary to buy up to date digital road maps from the supplier of the navigation system. They can be purchased directly by the producer of the digital maps – in many cases at a lower price [2].

[17] Recommended conditions from Siemens VDO. The conditions can vary from dealer to dealer.

a. Contracts with an unlimited activation period. The price includes the hardware and unlimited access to the German road map. Further C-IQ-based services can be purchased. The customer does not participate in the DVD/CD-Rom update service. If the customer does not buy any additional services the system works like a traditional navigation system.

b. Contracts with a limited activation period. The price includes the hardware and access to the German road map for two years. The customer participates in the update service and regularly receives free of charge DVD/CD-Roms with current data on it. When the activation time of the ordered content expires, the customer must purchase new services for further use of the system.

In the following, the pricing structure of the services with a limited activation period is analyzed more deeply. The customer can choose among the described services (see Section 3.2) and the period in which he wants to use each of them. The price depends on the combination of the ordered products and the duration of the contract. For each country or group of countries (like Spain/Portugal) the product assortment varies. For Germany, which has the most comprehensive assortment, the German digital maps from Tele Atlas or Navtech are available, plus ten individual travel products, one special product, three travel packages (each of them including four travel products) and 12 further bundles combining the German map with one or two travel products and some bundles additionally offer the Czech and/or the Slovakian map [6].[18] For all products and bundles the customer can choose among activation periods from one day up to two years. Different discounts are given if the customer orders a service for a complete month, one year or two years. Furthermore, for all single products (i.e. not in a bundle) prepaid cards for a period of five or twenty days are offered. Table 2 gives an example of the pricing system.

Because the prices for the other individual products, bundles and packages are similar, this example offers a good overview of the pricing system. The large number of products and the different measures of price discrimination lead to a very comprehensive and complex price matrix. The prices for all products – if they are available in the country – are the same all over Europe.

The change to the C-IQ business system has strong implications for the generation of revenue. Revenue streams from in-vehicle services can be segmented into two main categories: hardware devices and services revenues [28]. Traditional business models are based almost exclusively on revenues generated from hardware device. This implies that revenues were only generated once when the customer bought the system. There was little or no need to purchase a new device from the same brand. With the C-IQ system an important step has been made to generate additional revenues from services. It

[18] Some bundles are offered with and without TMC. As TMC is free of charge these combinations have not been counted. Identical bundles which are available both with Tele Atlas and Navtech maps have not been counted either.

Table 2. Prices in EUR for selected C-IQ travel products (C-IQ service center, [6])

	1 day	**1 mo.**	**1 year**	**2 years**	**Prepaid 5 days**	**Prepaid 20 days**
Road Map	2.99	19.99	89.00	159.00	12.99	39.90
Price steps	> 6 days	> 4 mo.	> 15 mo.			
Varta Guide	2.99	14.99	19.99	29.99	12.99	39.90
Price steps	> 5 days	> 1 mo.	> 1 year			
Michelin Guide	2.99	19.99	24.99	39.90	12.99	39.90
Price steps	> 6 days	> 1 mo.	> 1 year			
Road Map + Varta guide	4.99	29.99	99.00	169.00		
Price steps	> 6 days	> 3 mo.	> 14 mo.			
Sum prices of single products	5.98	34.98	108.99	188.99	not available	
Bundle savings	16.6%	14.3%	9.2%	11.0%		
Lifestyle Package	5.99	39.90	69.00	109.00		
Price steps	> 6 days	> 1 mo.	>1 year			
Sum prices of single products	11.96	59.96	89.96	139.69	not available	
Bundle savings	50.0%	33.5%	23.3%	22.0%		

is estimated that in the future service revenues will outpace hardware devices revenues [28]. Customers who have a contract with a limited activation period continuously generate revenue, independent of how long they use the hardware. Furthermore, C-IQ customers have a strong incentive to purchase the next hardware device from VDO Dayton. The implications of this shift to the new business models will discussed in more depth in the next section.

4 Implications Regarding the Change of the Business system

4.1 Economic Implications

Whenever an existing business system is changed the fundamental reason is to increase profits or, at the very least to assure current profits for the future. The underlying motives for the change could be many, including technical

progress which leads to product or process innovations, pressure from competitors and/or changing customer demand. In the following, two important motivations which are particularly interesting from an economic point of view are analyzed in more depth.

The analysis of the market model has identified one important motivation: driven by technical progress and the development of new services, new competitors are entering the market. Thereby the borders of the relevant market will fade away since the new competitors will offer navigation services combined with location-based services, personalized services etc., which in many cases will be offered on a "pay per use" basis (see Section 3.1).

Obviously a very strong new competitor for VDO Dayton will be the mobile phone operators. Cellular phones are widespread and they are able to offer user-centric navigation with dynamic rerouting, emergency call and roadside assistance, location based-services and further personalized services on a "pay per use" basis.[19] Furthermore, contractual relations and a billing infrastructure with their customers already exist [28, 49, 54]. However, further development is difficult to predict. One crucial question is, if and to whom will the car makers provide a connection to the vehicle computer system.[20] With access to the computer system supplier of aftermarket systems, even mobile systems could offer basic telematics services and brand-specific services. Brand-specific services are differentiated, exclusive services e.g. only for BMW or VW customers, presented on the screen in the respective corporate design. Until now, the offer of brand-specific services would require cooperation with the car makers, which is an advantage for the supplier of navigation systems, because they already have strong relationships with the main car makers. However, the introduction of an industry standard for the connection of telematics devices to vehicles would dramatically change the situation for all competitors [28].

A second important motivation has been identified through analysis of the revenue model: the possibility to generate additional revenue streams is mainly given by service revenues. In addition, the new indirect competitors will offer additional services as well. The challenge for VDO Dayton was not only to introduce new services but also to introduce an adequate pricing system for the billing of these services. VDO Dayton decided to offer the services on a "pay per use" basis, a (new) pricing system which the customers are not used to.

---

[19] More than three quarters of the Western European Population has a cellular phone. See [29]. i-mode customers can order maps and routes for two euros a month and the Michelin guide for 1 euro a month. Vodafone live! customers are offered maps and routes for 0.69 euro for two hours or 1.99 euro for one month and the Michelin guide for 0.39 euro for two hours or 0.99 euro for one month. See [7].

[20] This decision depends not only on economical considerations. The connection of devices from third parties to the vehicle computer system has strong IT security implications which must be tackled.

Strictly speaking the pricing system consists of a system of "time restricted flat rates". During the chosen activation period each service can be used as often as desired (i.e. the number of route requests during the activation period is unlimited), but for the customer the purchase of the content (the C-IQ-based services) has a "pay per use" character, especially when he orders services for a short period of time for a specific occasion, for example a Dutch road map for a one-day trip to the Dutch coast or a French road map and shopping guide for a one-week vacation in Paris.[21]

To maximize the revenues two types of price discrimination are applied: (1) for each product a discount is offered depending on the duration of the ordered activation period and (2) most of the products are also offered in a bundle form (mixed bundling) [34, 44]. The effectiveness of these strategies is influenced by the underlying cost structure. The higher the ratio of fix to variable costs is, the more effective these strategies are [34, 44]. The production of all C-IQ-based services leads to relatively high fix costs and low variable costs, since all data is produced once and stored digitally on CD- or DVD-Roms. The same applies to the additional distribution costs caused by the introduction of the new services (IT, service center, Internet presence).

The introduction of discounts is an incentive for the customer to buy a higher quantity (= longer duration) and thereby spend more money than he would normally have done without the discount scheme. If the variable costs are low this leads to increasing revenue. The success of bundling strategies essentially depends on the relation between the reservation price and variable costs (the higher the relation, the more successful the strategy).[22] The existence of relative low variable costs is again an important requirement for the success of this pricing strategy. Product bundling makes it possible to sell products which the customer would not have bought as a single product because in his view it is too expensive or simply because he is not familiar with the product. Bundling leads to higher revenues and could help to introduce (and sell) new products to the customer.[23] Furthermore, bundles give the customer the impression of buying something at a very fair price. As the price overview shows, the customer can enjoy high savings by buying the bundle

[21] In a strict sense the use of a navigation service must be defined as each request for a route made by the driver. But from the customer's point of view certain events with a determined duration and no interruptions in between, like day trips or vacations, are considered as one use of the system regardless of whether he makes one or 10 requests during the "use".

[22] The term "reservation price" derives from the auction markets. It is the price at which a person is willing to buy or sell a certain product. Bundling is only worthwhile if the variable costs are less than the reservation price [34].

[23] This is especially important for experience goods. Many C-IQ-based services belong to this category. In order to present its products, VDO Dayton offers its customers a free preview option with each DVD/CD-Rom update: two country maps and two travel guides can be chosen and tested for two days free of charge.

instead of single products. The customer has an incentive to spend additional money and a "positive image" effect is generated.

The product characteristics and the product assortment offer many options to introduce further types of price discrimination in the future [44]:

- The customer could be offered product bundles which he can configure on his own (self-customizing), e.g. three self selected travel products for a one-month period of use for 45 euros.
- Prices could be differentiated according to the time of use (intertemporal price discrimination), e.g. higher prices for travel guides during school breaks.
- Frequent users could be offered further discounts similar to frequent flyer schemes (premium program).
- Because the customer must register to use the C-IQ service and the purchase of each product is also registered, customers could be offered individual bundles and prices based on customer profiles (personalized services).
- The hardware and services could be combined in bundles.

However, it should be taken into account that the pricing matrix is already very complex and complexity will only increase with each new product. Therefore, the introduction of further measures of price discrimination could lead to customer confusion rather than higher customer satisfaction (and thereby higher revenues).

All offered C-IQ-based services are based on third party data. It is obvious that Siemens VDO has to pay for the integration of certain products like digital maps, because navigation is still the core application and an absolute must for in-vehicle service bundles, and only few companies offer digital maps [21]. However, with the integration of more and more well-known services from content providers with a high brand awareness, like the Michelin and Varta guides (and of course with a growing number of customers) the incentives grow for certain content providers to join the service portfolio and to pay for integration, i.e. to share in the revenues. For example, the integration of a restaurant finder for a fast food chain could lead to more visits to their restaurants, and if there is a co-marketing strategy, to a higher brand awareness. Therefore, a fast food chain would be willing to pay for the integration of their services.

With the introduction of the C-IQ system, Siemens VDO has not only made an important step to generate additional services revenue streams, but has also established a platform which allows the integration of a wide range of digital contents from third parties.[24] In the future, additional content providers could join the platform and offer their services. VDO Siemens provides the administration and billing infrastructure. Because the C-IQ system

[24] The introduction of hard-drive equipped navigation systems combined with fast data transmission will make it possible to offer (and also charge) songs or films, for instance. HD-based navigation systems are already offered in Japan. See: [26, 35].

allows the exact monitoring of to whom which products are sold, new models of payment/revenue sharing could be established based on the number or duration of use of the services. Therefore it is essential that the customer gets used to the "pay per use" pricing systems. The introduction of new pricing systems requires time and can only be made gradually. Siemens VDO has already taken the first step.

Another development could make it necessary to shift from hardware device revenue to services revenue. It is likely that the hardware device revenues will diminish: the hardware prices for navigation systems will decrease in the future (just like the prices for other electronic consumer goods) and it is probable that in the long run car makers will integrate factory-fitted navigation systems in all cars as a standard feature, like radios or air conditioners today [21]. Through this, the aftermarket volume for navigation systems will decrease significantly. Of course, sales to the car makers will rise at the same time, but margins will probably decrease and the brand could disappear. The C-IQ-based service platform could survive this process and link the customer to C-IQ products and the brand.

An additional future scenario is that Siemens VDO runs the system for a car maker or sells the complete business system to a car maker. In both cases basic telematics services could be integrated, as well as brand-specific services. All services are offered in the corporate design of the car maker. For the car maker, the system could be a very useful tool to stay in contact with its customers after the purchase of the vehicle, monitor his driving and consumer habits, promote and sell services and/or introduce new forms of a customer loyalty program. If Siemens VDO runs the system for a car maker distribution and sale is much easier, since it will be done through the car dealers. The dealers can influence the customer at the moment when he is buying the vehicle, which is probably the best moment to register the customer and sell additional in-vehicle services [28].

4.2 IT Security Implications

In the previous chapter, the economic implications of the change of the business model were discussed. As stated in the introduction, this change of the business model was only possible through the employment of IT security measures.[25] The importance of IT security as a prerequisite for the introduction of the C-IQ-based services is discussed in the following section and an outlook regarding what the introduction of further in-vehicle services will mean for the employment of IT security measures is provided.

IT security consists of two aspects: data security and data privacy. Data security is comprised of three basic requirements: (a) confidentiality – prevention

[25] In the English literature a difference is made between safety and security. Safety consists of protection against unintended incidents, whereas security means protection against intended attacks by an adversary [17]. In this article, only security aspects are tackled.

of unauthorized parties to capture, interpret or understand data, (b) availability – continuous and uninterrupted provision of services, and (c) integrity – the assurance that data have not been altered or manipulated by unauthorized parties [16, 47]. Several additional IT security requirements exist, depending on the specific environment. For the in-vehicle services discussed in this article, another important requirement is authentication – the assurance that the entity who is communicating is really who they claim to be [47].

Data privacy is comprised of the protection of personally related data. In most developed countries the protection of personal data is protected by data protection laws. In Germany, for instance, privacy is guaranteed by the constitution and with several laws. Therefore, data protection laws must always be taken into account when new business models are developed which include the use of personal data.

In traditional business systems for navigation systems (see Section 3.6), IT security was of relatively little importance. It was practically impossible to identify the user of the navigation system and the vehicle could not be located in order to track its movements and create a user profile. There was only one major aspect which approached IT security requirements. When traditional navigation systems entered the market, the navigation software on the CD (digital maps etc.) was not encrypted, i.e. confidentiality of the data was not provided. Anyone who had a CD writer could easily replicate the CD and use it in his own navigation system. Consequently, owners of navigation systems attempted to obtain unauthorized copies with updated map data. Although, generally speaking, the loss from CD or DVD piracy causes a significant decline in profits in the affected industries (such as the music industry), the impact on the supplier of navigation systems was relatively weak, since revenues were traditionally generated mainly through hardware sales [9].

With the introduction of the C-IQ business model the situation changed substantially. As described, revenues are shifting from hardware to services revenue. Independent of the amount of services the user orders, all services (all european maps, all guides etc., see Section 3.2) are already stored digitally on the CDs or DVDs when customers purchase the navigation system. Additionally, customers who have contracts with a limited activation period regularly receive DVD/CD-Roms with the current data on it free of charge. Therefore, confidentiality is an absolute must to run the C-IQ business system. Without data encryption, customers could use all the services without being charged for them. Yet here, IT security requirements are much more comprehensive and complicated than in the case of music CDs or DVDs, where IT security measures (encryption) only have to be applied to avoid unauthorized access. Because the C-IQ-based services are offered on a "pay per use" basis, the customer must be offered conditional access to the protected data on the CDs/DVDs. This means that it must be possible to control (a) which data (content) can be accessed, (b) the period of access or the number of uses of a determined service, (c) to whom the access is given, and (d) whose specific device can access the information. IT security measures must be applied to

enforce these complex rules [57]. The management of rules for digital content is named Digital Rights Management (DRM) [57].[26]

To implement these rules within the C-IQ business system interaction between the customer, his or her navigation system, and the C-IQ service center is required, and the customer's identification and the hardware being used is needed. The customer identification takes place when the customer registers for the first time with the C-IQ service (see Section 3.4). The registration process requires not only personal details (name, address, preferred payment method) but also details of the purchased navigation system including the navigation ID (navi ID). Each navigation system has its own unique ID. When the customer orders new services, a PIN code is generated and transmitted to the customer (e.g. via SMS), who enters it into his navigation system. This code contains, among other things, information about the ordered services (e.g. a French road map), the period of use and the navigation system for which the code is intended. [27]

During this procedure it must be assured (1) that the message cannot be altered, for example to extend the activation period (integrity) and (2) that the code functions only with the navigation system for which it is intended in order to prevent services on other navigation systems activated by the same code without paying for it (authentication).

To fulfill the requirements of integrity and authentication a message authentication code (MAC) (= cryptographic checksum) can be employed [47].[28] This technique requires that the two communicating parties (the navigation system and the computer system of the Siemens VDO service center) share a common secret key K. When the service center wants to send a message (e.g. access to the French road map for two weeks) to the customer, it calculates the MAC as a function of the message and the key. The message and the MAC are transmitted to the customer, who enters it into his navigation system (i.e. the PIN code consists of two parts: the message with the desired services plus the navi ID and the MAC). The navigation system performs the same calculation on the received message, using the same secret key, to generate a new MAC. The transmitted MAC from the service center is compared to the calculated MAC.

If the secret key is known only by the receiver and the sender, and if the received MAC matches the calculated MAC, then the receiver is assured

[26] For a general overview of Digital Rights Management see [8].

[27] If customers wish to cancel ordered services an additional step is needed. First, the customer must request a revocation code for the service to be cancelled. After this code has been entered into the navigation system, the system will generate a confirmation code. The customer must contact the C-IQ service center or transmit the confirmation code via the Internet. The confirmation code is needed to control that the customer really has deactivated the content. After the transmission of the confirmation code reimbursement of the residual value can be made [6].

[28] This is not necessarily the solution applied by Siemens VDO, but it is one viable solution to meet the described requirements.

that the message has not been altered (integrity), and that the message is from the intended sender. No one else could prepare a message with a proper MAC without knowing the secret key (authenticity). Since for each navigation system a unique secret key is applied, customers or third persons cannot use the code to activate content on other navigation systems.

At this stage, the message as a whole is still transmitted in the clear (plain text). To provide confidentiality the MAC, which is calculated with the message as input, can be connected to the message and the entire block (the PIN code) is then encrypted [47]. This requires a second secret key, which is also shared by the sender and receiver.[29]

The problem of key distribution, which is a very important aspect to consider whenever symmetric encryption is applied, is relatively easy to deal with in this case, since Siemens VDO can implement the secret keys in the navigation system before delivering it to the dealers [47]. When the customer orders some content, the service center can select the respective secret keys with the help of its database, which stores the details of the customer, the navi ID and the corresponding secret keys.

It is easy to recognize that IT security is an absolute prerequisite to run the C-IQ business system. With further development of the system, mainly driven by technical progress, IT security will become even more important. As discussed in Section 4.1, with the C-IQ system, Siemens VDO has established a platform which allows the integration of a wide range of digital content on a conditional access basis. Within the next few years, navigation systems will move away from CD- and DVD-based players to hard-drive based players. Models shown at the CES in Las Vegas in 2005 already had a 20 GB hard drive, USB ports and slots for removable media [26, 35]. This significant rise in performance will allow users to integrate a wider range of digital content, which can be uploaded through the unit's USB port or via a wireless connection [31]. It is assumed that one core application will be MP3 storage and playback. Additionally, it is only a matter of time until video storage and playback are also possible. The integration of digital content from third parties over a USB port or a wireless connection will require a more comprehensive Digital Rights Management system than described here, which must include the content providers as well. Cooperation partners could, for instance, offer digital content to be downloaded from their homepages for exclusive use with a C-IQ navigation system.

Digital Rights Management systems in vehicles are difficult to implement, because they are bound by certain restrictions and have special requirements. Compared with a notebook or a desktop computer, the processor performance is weaker and there is less memory capacity. Furthermore, a Digital Rights Management system in a vehicle now must be realized with very little external

[29] Actually two independent secret keys are not needed. With the help of key derivation two keys K1 and K2 can be generated from one secret master key K (K1 = g(K), K2 = h(K)).

communication. Until now, the only communication interface has been with the user [57].[30] The best solution from the IT security point of view would be a Digital Rights Management system based on a Trusted-Computing solution. A Trusted-Computing solution includes a hardware security module, called a Trusted Platform Module for the protection of cryptographic keys, the trusted employment of symmetric and asymmetric cryptographic functions and the integration of a real physical random numbers generator [57].

This analysis has tackled the central IT security aspects which emerged from the introduction of the new C-IQ business system. There are further IT security requirements which must be taken into consideration, such as communication security, e.g. for the transmission of confidential data or electronic payment, and, as previously mentioned, privacy for the protection of personal data [30, 55]. In addition to this it must be taken into account that IT security requirements are always placed between the conflicting goals of usability and profitability, since a higher security level normally leads to higher production costs.[31] A sufficient level of usability is important not only for the acceptance of services by the customer, but also for safety reasons. To meet all requirements discussed, it is important to consider IT security even during the design of new in-vehicle services [51].

5 Conclusion

In this article, an innovative business system for navigation systems and location-based services was presented and the most important economic and IT security implications were analyzed. For the first time, an aftermarket navigation systems supplier has introduced a business system which not only is based on the generation of hardware devices revenue, but is also based on generating services revenue.

The analysis shows that IT security can be seen as a prerequisite for the introduction of the new business system and for the shift from hardware devices to services revenue. Since the core services are all stored digitally on CDs or DVDs and conditional access was introduced, the content had to be protected from piracy and complex usage rules had to be established. Both were possible only with the help of advanced IT security measures. With the introduction of further in-vehicle services and the connection of the navigation system to the vehicle computer system (see Section 4.1) the importance of IT security will grow even more. Therefore, it is not an exaggeration to state that IT security enables the introduction of innovative in-vehicle services.

From an economic point of view, the new system has some obvious advantages and offers opportunities for the future: Among them are (1) new

[30] For further requirements and restrictions for Digital Rights Management systems in vehicles see [57].

[31] This is one reason why Trusted-Computing solutions are not employed in navigation systems.

revenue streams can be generated by offering a wide range of additional services which can be ordered in a very flexible way; (2) the customer gets used to a "pay per use" pricing system which is crucial for the further development and sale of new services; (3) with customer registration, a user history of the services ordered and the billing system allows the introduction of personalized services and new payment schemes not only for the customer but also for content providers or other cooperation partners; and (4) brand awareness and a higher customer loyalty are generated.

Despite the apparent advantages there are some factors which could undermine the success of the C-IQ system: (1) the system is difficult to understand because of the different types of contracts, the large product assortment with many different bundles, and the comprehensive and complex pricing matrix; and (2) in the past, willingness to pay for in-vehicle services was low [10, 20, 58, 28]. However, in the past, a subscription was necessary for most of the offered services and the customer had to pay an additional monthly bill. The C-IQ system could overcome the customer's aversion of being billed once a month by offering the services on a "pay per use" basis [20, 28, 54]. (3) Currently the major threat for the success of the C-IQ system, and for all other traditional navigation systems, seems to be the entrance of new competitors who offer mobile devices for *user-centric* navigation and additional *in- and outside-vehicle* services.

In summary, the market for in-vehicle services is characterized by significant uncertainties. Major influential factors are: the technical development (including the development of IT security solutions); the acceptance of services by the customer; his willingness to pay for these services; and safety issues. Safety issues should not be ignored, since driving is a demanding task which requires constant concentration and appropriate maneuvers of a vehicle on the road. Therefore, the introduction of in-vehicle services must be carried out with utmost care and it should be guaranteed that the driver always has his "eyes on the road and hands on the wheel".

References

1. ADAC – Auto News – Kleines Multitalent. ADAC Motorwelt, 2004.
2. ADAC – Kaufberatung und Tipps (Erstausrüstung, Nachrüstung). ADAC, July 2004. `www.adac.de`.
3. ADAC – Verkehr, Eckdaten, August 2004. `www.adac.de`.
4. ADAC – Praxistest 2003 – Navigationssysteme zum Nachrüsten, December 2003. `www.adac.de`.
5. ADAC – Praxistest 2003: Navigationsgeräte – Alle getesteten Systeme, July 2003. `www.adac.de`.
6. C-IQ: Information, July 2004. `http://c-iq.vdodayton.com`.
7. ViaMichelin, August 2004. `www.viamichelin.de`.
8. Eberhard Becker, Willms Buhse, Dirk Günnewig, and Niels Rump, editors. *Digital Rights Management, Technological, Economic, Legal and Political Aspects*. Springer-Verlag, Berlin/Heidelberg, 2003.

9. Larry Boden. CD-Rom, Piracy and the emerging Technology Fix. *CD-ROM Professional*, 8(9):68–80, September 1995.
10. Stephan Buse. Der mobile Erfolg. Ergebnisse einer empirischen Untersuchung in ausgewaehlten Branchen, In: Keuper, Frank (Hrsg.), *Electronic Business und Mobile Business. Ansaetze, Konzepte und Geschaeftsmodelle*, pages 89–116. Wiesbaden, Gabler Verlag, 2002.
11. Edmund Chew. Siemens VDO hits turnaround goal for '03. *Automotive News*, 78(6071):30, December 2003.
12. Edmund Chew. Supplier takes lead role in pushing navigation systems. *Automotive News Europe*, 8(14):17, July 2003.
13. David Crawford. Traffic information systems. *Automotive Engineer*, 24:34–47, June 1999.
14. VDO Dayton. Navigation, July 2004. `www.vdodayton.com`.
15. Marcus Efler. Firlefanz fliegt raus. *Focus*, 21:110–111, 2004.
16. BITKOM e.V. Sicherheit für Systeme und Netze in Unternehmen – Einführung in die Problematik und Leitfaden für erste Maßnahmen, August 2002. `www.bitkom.org`.
17. Hannes Federrath and Andreas Pfitzmann. Gliederung und Systematisierung von Schutzzielen in IT-Systemen. *Datenschutz und Datensicherheit (DuD)*, 12:704–710, 2002.
18. Mark Fischetti. Getting There. *Scientific American*, 286(5):42–43, March 2002.
19. TMC Forum. What is Traffic Message Channel (TMC). TMC Forum, June 2004. `www.tmcforum.com/en/about_tmc/what_is_tmc/what_is_tmc.htm`.
20. Kilian Frühauf and Rainer Oberbauer. Web in the car – Mobile Commerce als Herausforderung für Automobilhersteller, In: Günter Silberer, Jens Wohlfahrt, and Torsten Wilhelm, (Hrsg.), *Mobile Commerce. Grundlagen, Geschäftsmodelle, Erfolgsfaktoren*, pages 381–398, Wiesbaden, Gabler Verlag, 2002.
21. Laura Clark Geist. Future GPS could adapt performance to roads. *Automotive News*, 78(6080):22, February 2004.
22. GEO. Zoom: Stau Räume. *Geo Magazin – Hatschepsut*, July 2004. `www.geo.de`.
23. Martin Gersch. Cooperation as instrument of competence management. *International Journal of Management and Decision Making (IJMDM)*, 4(2–3):210–229, 2003.
24. Martin Gersch. Versandapotheken in Deutschland – Die Geburt einer neuen Dienstleistung – Wer wird eigentlich der Vater? *Marketing ZFP (Sonderheft Dienstleistungsmarketing)*, 26:59–70, 2004.
25. Amy Gilroy. Car Navigation Cues Up With Real-time Traffic Info. *TWICE – This Week in Consumer Electronics*, 16(12):24, May 2001.
26. Amy Gilroy. Navigation Shifts to Hard Drive Models. *TWICE – This week in consumer electronics*, 20(1):122, January 2005.
27. Hans Robert Hansen and Gustaf Neumann. *Wirtschaftsinformatik I. Grundlagen betrieblicher Informationsverarbeitung*, 8th ed., Stuttgart, Lucius & Lucius Verlagsgesellschaft, 2002.
28. Michael Heidingsfelder et al. Telematics: How to hit a moving target – A roadmap to success in the Telematics arena, June 2004. `www.rolandberger.de/documents/2340078/RB_Telematics_How_to_hit_a_moving_target_A_roadma_2001.pdf`.
29. TNS Infratest. Monitoring Informationswirtschaft, 7. Faktenbericht 2004. TNS Infratest, München, August 2004. `www.nfo-bi.com/bmwa`.

30. Thilo Koslowski. Opportunities and challenges in the telematics industry. Presentation at the conference ESCAR 2004 – Embedded Security in Cars, November 2004.
31. Stacy Lawrence. Wireless on Wheels – Carmakers are taking telematics to the streets. *Technology Review*, 108(1):22–23, January 2005.
32. Franz Lehner. Lokalisierungstechniken und Location Based Services. *WISU*, 33(2):211–219, 2004.
33. Marie McMorrow. Telematics – exploiting its potential. *Manufacturing Engineer*, 83(1):46–48, February 2004.
34. Torsten Olderog and Bernd Skiera. The Benefits of Bundling Strategies. *Schmalenbachs Business Review*, 52(2):137–159, 2000.
35. Joseph Palenchar. Pioneer Using Hard-Drives With DVD-Recorders, Car Navigation. *TWICE – This Week in Consumer Electronics*, 18(1):60–61, January 2003.
36. Michael Rappa. Managing the Digital Enterprise: Business Models on the Web, April 2004. `http://ecommerce.ncsu.edu/topics/models/models.html`.
37. Jahn Rentmeister and Stefan Klein. Geschäftsmodelle in der new economy. *WISU*, 30(3):354–361, 2001.
38. E. A. G. Robinson. *Monopoly*, The Cambridge Economic Handbooks, Cambridge, James Nisbet & Co. Ltd., 1963.
39. Joan Robinson. *The Economics of Imperfect Competition*. London, Macmillan & Co. Ltd., 2nd ed., 1964.
40. Siemens. Annual report 2003, July 2003. `www.siemens.com/Daten/siecom/HQ/CC/Internet/CC_Unitwide/WORKAREA/gbericht/templatedata/English/file/binary/000_GB2003_E_1129103.PDF`.
41. Siemens VDO Automotive. About us. July 2004, `www.siemensvdo.com`.
42. Siemens VDO Automotive. Products, Solution & Services, July 2004. `www.siemensvdo.com`.
43. Siemens VDO Automotive. Worldwide, July 2004. `www.siemensvdo.com`.
44. Bernd Skiera. Preispolitik und Electronic Commerce. Preisdifferenzierung im Internet, In: Wamser, Christoph (Hrsg.), *Electronic Commerce – Grundlagen und Perspektiven*, pages 117–130, München, Vahlen Verlag, 2000.
45. Michael Spehr. Der schnellste Routenführer der Welt. VDO Dayton bietet mit dem MS 5500 Spitzentechnik und viel Tempo. *Frankfurter Allgemeine Zeitung*, September 26 2002.
46. Michael Spehr. Die Navigations-DVD gibt es gratis. Bsei VDO Dayton bezahlt man nur für die gebuchten Teilinformationen. *Frankfurter Allgemeine Zeitung*, September 24 2002.
47. William Stallings. *Cryptography and Network Security – Principles and Practices*. New Jersey, Prentice Hall, 3rd ed., 2003.
48. OC & C Strategy Consultants. Die M-Commerce-Strategien deutscher Großunternehmen. Eine empirische Studie von OC & C Strategy Consultants, December 2000.
49. Telenav. GPS Navigation Systems Feature Speech Interface. Audiotex Update, February 2003.
50. test. Navigationsgeräte Test – Wegweisend. *Test, No. 1*, pages 67–70, 2002.
51. O. Tettero, D. J. Out, H. M. Franken, and J. Schot. Information security embedded in the design of telematics systems. *Computers & Security*, 16(2):145–164, 1997.

52. Paul Timmers. Business models for electronic markets. *Focus theme*, 8(2):3–8, 1998.
53. Paul Timmers. *Electronic Commerce. Strategies and Models for Business-to-Business Trading.* Chichester, Wiley & Sons Ltd., 1999.
54. Richard Truett. Telematics execs seek a new route. *Automotive News*, 77(6009):1–2, October 2002.
55. André Weimerskirch and Christoph Paar. IT-Security in Geoinformation Systems. *Geoinformation Systems*, pages 1–7, April 2005. `www.geoinformatics.com/freedownloads/itsecurity.pdf`.
56. Bernd W. Wirtz. *Electronic Business.* Wiesbaden, Gabler Verlag, 2nd ed., 2001.
57. Marko Wolf, André Weimerskirch, and Christoph Paar. Digitale Rechteverwaltung – Unerlaubte Vervielfältigung digitaler Inhalte verhindern und neue geschäftsmodelle absichern. *Elektronik Automotive*, 2:44–48, April 2005. `www.escrypt.org/download/Digitale_Rechteverwaltung.pdf`.
58. GPS World. Consumer Telematics Attitudes Gauged. *GPS World*, 15(3):47, March 2004.
59. Chris Wright. DVD navigation has rough road in U.S. *Automotive News*, 78(6067):34, November 2003.