Access Control With High Security Credentials

Michael Kapusta

Nicklas Lindstrom

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
ISRN LUTFD2/TFRT--6003--SE
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2016

Abstract

Developing security regardless of its format is a constant cat and mouse game were adversaries are either in the midst of trying to crack your solution, or they may have already cracked it. A cryptographic algorithm may be unfeasible to crack from a mathematical perspective but as long as a human being is the one developing the solution, a human error is always possible.

A large quantity of the current security solutions on the Physical Access Control Systems market are, as will be shown in this thesis, riddled with human errors. Security systems that are portrayed by their developers as secure even though they are not, give the users a false sense of security. The insecure Physical Access Control Systems are, as will be shown in this thesis, most frequently a result of proprietary solutions by the developers.

The thesis analyzes and evaluates various authentication and authorization techniques with a high level of security for smart cards and smartphones, within the scope of Physical Access Control Systems. This includes an analysis of standards and protocols such as PIV, PLAID, FICAM and FIPS 201 with respect to their cryptographic properties, workflows and user management. The thesis also includes prototyping of such functionality on an embedded system in combination with a smartphone.

Acknowledgment

We would like to extend our gratitude to the following persons:

Mathias Bruce - *Supervisor and mentor at Axis Communications, Lund*

Johan Adolfsson - *Supervisor and mentor at Axis Communications, Lund*

Henrik Fredriksson - *Employee at Axis Communications, Lund*

Martina Maggio - *Supervisor and mentor at Department of Automatic Control, Faculty of Engineering, Lund University*

Karl-Erik Årzén - *Examiner at Department of Automatic Control, Faculty of Engineering, Lund University*

We would also like to thank the employees at Axis Communications in Lund for aiding us with both the equipment and the expertise needed to realize this master thesis.

Contents

Abbreviations

AES Advanced Encryption Standard

AID Application Identifier

APDU Application Protocol Data Unit

ARM Advanced RISC Machine

CBC Cipher Block Chaining

CIV Commercial Identity Verification

CPU Central Processing Unit

CRL Certificate Revocation List

DES Data Encryption Standard

DSA Digital Signature Algorithm

EAC Electronic Access Control

EAL Evaluation Assurance Level

ECB Electronic Codebook

ECC Elliptic Curve Cryptography

EIA Electronics Industry Association

FASC-N Federal Agency Smart Credential Number

FICAM Federal Identity Credential and Access Management

FIPS 201 Federal Information Processing Standard Publication 201

FTP File Transfer Protocol

HSM Hardware Security Module

HSPD-12 Homeland Security Presidential Directive 12

HTTP Hypertext Transfer Protocol

ICAM Identity Credential and Access Management

ICC Integrated Circuit Card

IP Internet Protocol

ISO International Organization for Standardization

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

LSB Least Significant Bit

MDB Multidrop Bus

MSB Most Significant Bit

NFC Near Field Communication

NIST National Institute of Standards and Technology

NSA National Security Agency

OSDP Open Supervised Device Protocol

OSI Open Systems Interconnection

PACS Physical Access Control System

PIV Personal Identity Verification

PIV-C Personal Identity Verification Compatible

PIV-I Personal Identity Verification Interoperability

PKI Public Key Infrastructure

PLAID Protocol for Lightweight Authentication of Identity

PoE Power over Ethernet

RAM Random Access Memory

RSA Rivest Shamir Adleman

SC Side-channel

SCA Smart Card Alliance

SD Secure Digital

SIA Security Industry Association

SSL Secure Socket Layer

UICC Universal Integrated Circuit Card

USB Universal Serial Bus

UUID Universal Unique Identifier

1 Introduction

Axis Communications is a Swedish-based company founded in 1984 and is currently considered the market leader in network video and surveillance cameras [26]. Axis products are used in public places and areas such as airports, motorways, trains, casinos and banks. More information about Axis Communications and their products can be found at their website www.axis.com.

1.1 Background

The Network Door Controller A1001 is a distributed access system that Axis Communications has developed. A user has the possibility to connect a card reader to the A1001.

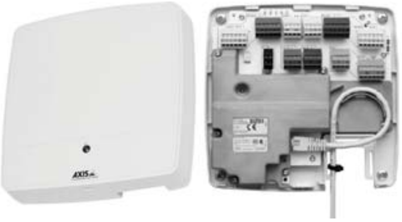


Figure 1: Axis A1001 Network Door Controller.

There is no established industry standard for authorization and authentication and one can find many solutions, each of them having different security properties. In some cases, communication between cards and card readers are secured by symmetrical cryptography or not secured at all, ditto the communication between the reader and the control unit. In other cases a combination of asymmetric and symmetric cryptography is used and sometimes communication is secured end-to-end between a smart card and a controller.

1.2 Project goal

The goal of this work is to analyze and evaluate various authentication and authorization techniques with a high level of security for smart cards and other mobile devices. This includes an analysis of standards and protocols such as PIV, PLAID, FICAM and FIPS 201 with respect to the cryptographic properties, workflows, user management, etc. This work also includes the prototyping of such functionality on the embedded system A1001 and on a smartphone.

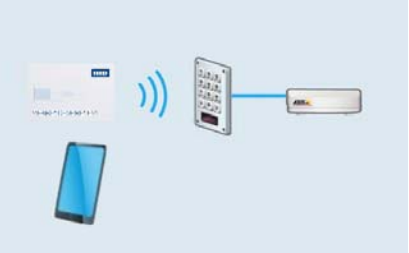


Figure 2: Overview of the system that this thesis will conduct work upon.

1.3 Outline

All the theory and fundamentals used in this thesis are described in Chapter 2. In Chapter 3 an evaluation of the existing solutions and suggested improvements is done based on the theory in Chapter 2. Chapter 4 covers all parts (both hardware and software) of the implementation. In Chapter 5 the results of the implementation are presented. Chapter 6 contains discussion of what this thesis has led to. Chapter 7 contains suggestions for future work based on this master thesis.

1.4 Individual contributions

Most of the work of this thesis has been divided equally between the two authors, Michael Kapusta and Nicklas Lindstrom. However, Nicklas Lindstrom has done most of the software on the embedded side whilst Michael Kapusta has done most of the development on the Android side of the prototype.

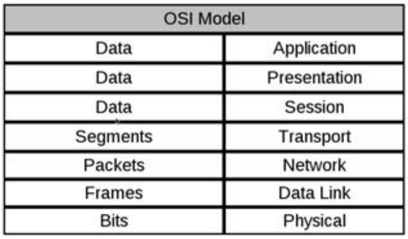
2 Theory

Chapter 2 contains all the theory and technical information which are needed for the evaluation in Chapter 3.

2.1 Fundamentals

2.1.1 OSI Model

The International Organization for Standardization (ISO) states that the Open Systems Interconnection (OSI) model enables exchange of information between a set of one or more computers. The OSI model uses abstract models (layers) to achieve this, where the interconnections between these models are described by their external behavior [41, p. 4]. The OSI-model does work well as a reference model for explaining how communication works [59]. The different layers of the OSI model and what they transmit are shown in Figure 3.



OSI Model	
Data	Application
Data	Presentation
Data	Session
Segments	Transport
Packets	Network
Frames	Data Link
Bits	Physical

Figure 3: The OSI Model.

The physical layer contains rules about how the sending of data over a transmission medium should be conducted. These rules involve transmission speed, synchronization, and how binary numbers should be represented when transmitted [59]. The data link layer makes sure that communication between two adjacent nodes in a network is correct by framing the data [59, p. 118]. The network layer contains rules for sending data over one or more networks. By using network addresses the layer enables data to be transported to the correct receiver. The transport layer makes sure that data from different applications is correctly received by the receiving end using flow control, data stream support, and error checking schemes. The session layer contains rules for the synchronization when two computers take part in a session. The presentation layer translates the data it receives, thus it contains rules for coding the information so it can be interpreted correctly [59, p. 119]. The application layer enables both the user and the computer-software to access the network.

2.1.2 TCP/IP Model

Compared to the OSI model, the top three layers (application, presentation and session) are combined into one in the TCP/IP model [59, p.126]. This means that an internet application itself is responsible for representing the ones and zeros, and possible encryption. Also, the bottom two layers (link and physical) are combined into one. This means that the operator has the freedom to choose the underlying data network which transports the data.

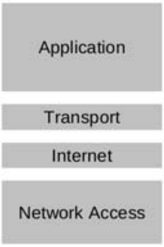


Figure 4: The TCP/IP Model.

2.1.3 Duplex

When a communications channel only allows transmissions in one direction between its endpoints, it is called simplex [49]. A half duplex channel allows for communications in both directions between two endpoints. The endpoints share a joint channel and only one endpoint can use the channel at a given time. A full duplex channel allows for communications in both directions betwen two endpoints. The endpoints in a full duplex connection either share a joint half duplex channel that allows bidirectional transmissions or they each have their separate channel. This enables the sending of data simultaneously between the endpoints.

2.1.4 Alice, Eve and Bob

The usage of the names Alice, Eve and Bob in cryptography is a simplification of labeling the different parties that may take part in an exchange of information [91, p. 2]. Alice and Bob are the parties that share the information between them, whilst Eve has the goal of eavesdropping on the communication.

2.1.5 Confusion

The definition of confusion in cryptography is making the relationship between the message and the encrypted message a very complex one [87, p. 54]. Attaining a complex relationship is done by pushing the message through non-linear functions.

2.1.6 Diffusion

The definition of diffusion in cryptography is the avoidance of recurring structures in the message that is encrypted [87, p. 53]. This is done by diluting the encrypted message thus avoiding redundancy in the message. An attacker will then have issues gaining an understanding of the structure of the encrypted message.

2.1.7 Near Field Communication

The first version of Near Field Communication (NFC) came out in 2003. NFC works as a data transportation method that is built upon short wave radio technology at a frequency of 13.56 MHz [9, p. 144]. A device that uses NFC is either passive or active, where an active device has the possibility to send and read from other devices whilst passive devices can only be read [66].

The Gartner Hype cycle model can be used to separate the hype of a new technology from its commercial potentials [38]. The Hype cycle model is graphically shown as a curve as in Figure 5. A product can be placed in five different phases in the curve as explained in Figure 6. In 2014 NFC was placed in the disillusionment state, with the expectation of crossing in to the plateau stage between 2016-2019 [9, p. 146]. This development will be the result of standardization and commitment from smartphone companies [9, p. 158].

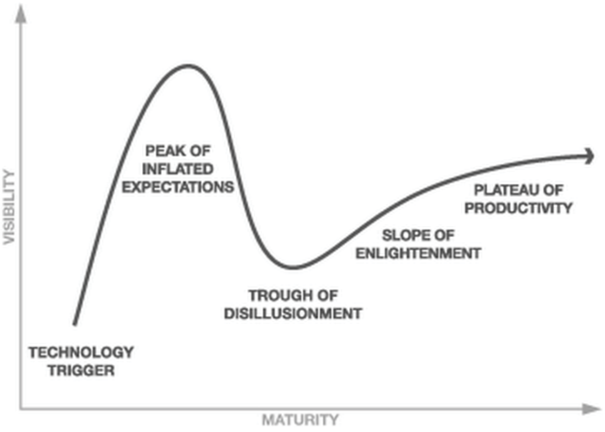
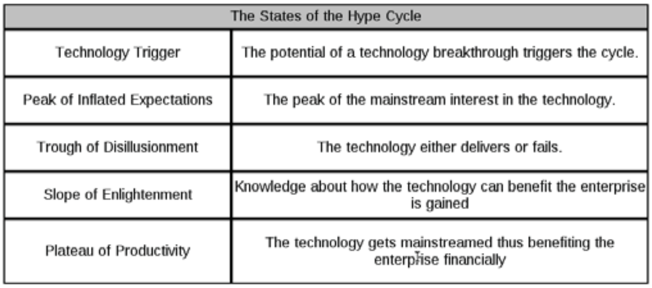


Figure 5: The Gartner Hype Cycle [38].



The States of the Hype Cycle	
Technology Trigger	The potential of a technology breakthrough triggers the cycle.
Peak of Inflated Expectations	The peak of the mainstream interest in the technology.
Trough of Disillusionment	The technology either delivers or fails.
Slope of Enlightenment	Knowledge about how the technology can benefit the enterprise is gained
Plateau of Productivity	The technology gets mainstreamed thus benefiting the enterprise financially

Figure 6: The States of the Hype Cycle [38].

2.1.8 Hardware Security Module

The Hardware Security Module (HSM) is used to protect data such as cryptographic keys and other sensitive information [89, p. 3]. The HSM provides cryptographic functions, one of these functions is the possibility of sampling a noise generator that is a part of the HSM thus creating a hardware based random generator [89, p. 7]. The HSM uses a number of different physical security measures to protect itself against physical attacks [89, p. 8]. Tamper resistance is achieved by making the HSM physically hard to break in to, for example by strengthening the outer shell. Tamper evidence is achieved by making it evident when the HSM has been attacked physically for example by adding a unique seal that is hard to replicate. The HSM can be designed to detect tamper attempts for example by monitoring the outer shells electrical properties. The HSM may be designed so that it wipes all data within it, when it detects tampering. Military HSM take it a step further using thermite explosives to self destruct when tampering is detected.

2.2 Attack Methods on Physical Access Control System

This chapter goes through well-known attack methods on Physical Access Control System (PACS).

2.2.1 ID-Leakage

If part of the data is constant each time the card reader authenticates a certain Integrated Circuit Card (ICC) Eve will then be able to identify the ICC [68, p. 7]. If not protected correctly the ICC can reveal too much information about itself when being scanned by the card reader, a result of being badly designed from a security perspective [47]. For exemple if the ICC shares its ID number when asked by a card reader to do so, instead of first authenticating the card reader, then an ID-Leakage takes place.

2.2.2 Man In The Middle Attack

By placing an emulator between the Card Reader and the ICC, Eve can modify the data maliciously without being detected [68, p. 7]. This attack is referred to as a Man In The Middle Attack (MITM).

2.2.3 Replay attack

Symmetrical cryptographic protocols can be vulnerable to attacks where the host is fooled to accept a copy of a message that was previously sent. Using a copy of a message that was previously sent, Eve may gain access to the system [68, p. 7]. This attack is refereed to as a reply attack since Eve ”replays” past messages.

2.2.4 Side-channel attack

When Eve exploits the traditional cryptographic model she attacks the mathematics behind the cryptographic scheme as shown in Figure 7. Side-Channel

(SC) attacks are attacks that target implementation weaknesses in the cryptographic algorithms [94, p. 2]. The SC attacks utilize the correlation of computation and difference in physical and digital measurements as shown in Figure 8. A large number of these attacks havebeen shown to be optimized using stochastic methods and decision theory [94, p. 22].

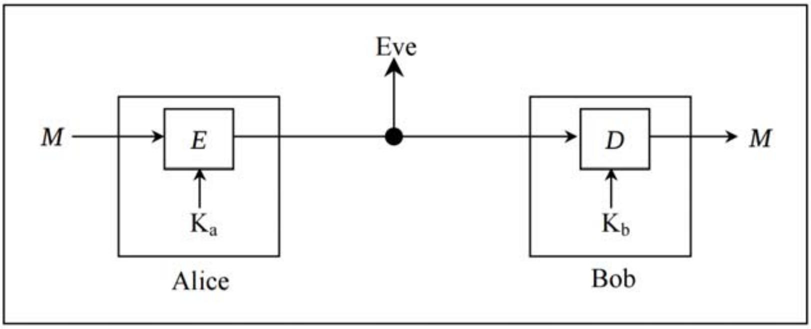


Figure 7: The traditional cryptographic model [94, p. 6].

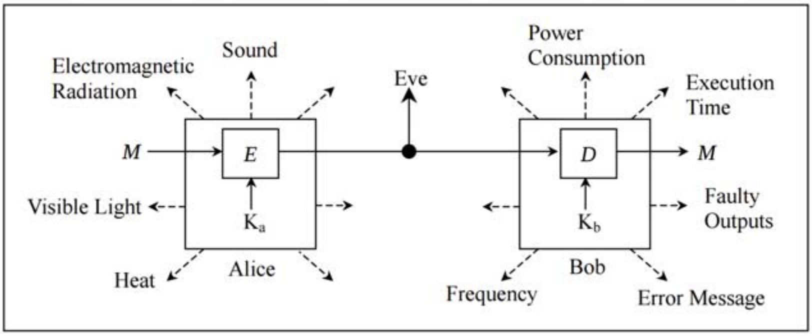


Figure 8: The cryptographic model including side-channel [94, p. 6].

By analyzing the execution time of a cryptographic operation, Eve can gain information about the keys being used [94, p. 11]. A fault attack is when Eve induces faults in the system, thus effecting the system to behave as Eve wants it to behave [94, p. 13]. By analyzing the power consumption of the cryptographic device Eve gains knowledge about cryptographic operations in the system [94, p. 15]. Cryptographic devices exude electromagnetic radiation, this can be used to understand the underlying relationship within the hardware [94, p. 17]. Correlations between computations and the sound of a Cryptographic device processor has been proven to be a factor. Thus this can be used in an acoustic attack [94, p. 18]. An example of an error message attack is the Padding Oracle attack which is described in Chapter 2.2.5. Cache-based attacks utilize cache misses which causes a delay that gives information about underlying relationships within the hardware [94, p. 20]. Frequency-based attacks utilize that certain devices (for example smartphones) use frequency to communicate [94, p. 21]. Strategies to combat SC attacks involve:

1: Introducing arbitrary timing elements in the execution process, thus making the individual run times unique.
2: Replacing the assembler's critical instructions with ones that makes the process of analysis a lot tougher for Eve. Alternatively one can re-engineer that circuitry parts the handle memory transfers and arithmetics.
3: Masking the data and keys that are used with random values that are unique for each run.

2.2.5 Padding Oracle attack

The padding oracle attack is used to break a Cipher Block Chaining (CBC) encryption [52]. A method of padding CBC is using PKCS7, each padding byte that is added contains the total number of bytes being added as shown in Table 1.

01		
02	02	
03	03	03

Table 1: PKCS7 padding structure.

The padding bytes are not valid if they contain the wrong number. This will generate a error exception which uncaught will enable the attack. Eve simply has to send random data cipher texts with different variations of padding until there is a match that does not generate the error exception. Eve will then have enough information to decrypt all the information that is sent between Alice and Bob with the CBC scheme.

2.2.6 Forward Secrecy Compromisation

To obtain forward secrecy all of the keys that are used in a cryptographic system should be uniquely generated [54]. Thus if one key is compromised none of the other keys will be compromised. This is especially a vulnerability during the session key exchange. A Diffie-Hellman key exchange with unique keys for each new session is an example that enables forward secrecy.

2.3 Transport protocols

This chapter will go through different transport protocols that will be evaluated in the thesis. The transport protocols are between the card reader and the Axis A1001 Network Door Controller.

2.3.1 Wiegand

According to HID Global, a company that manufactures security solutions, the term Wiegand is used to describe several different aspects that relate to cards and access control readers. One of these is the reader-to-control interface [43, p. 1]. The reader-to-control interface (physical layer) consist of three conducting wires "D0", "D1", and "DR" where "D" stands for data and "R" for return. When the reader receives binary data from a card it sends it to the controller through the conductors. If no transmission is in progress both D0 and D1 are set to one, the physical representation of this is that they are set to the voltage 5 VDC. Zeros are sent by setting D0 to zero (0 VDC) and D1 to one (5 VDC). Ones are sent by setting D0 to one (5 VDC) and D1 to zero (0 VDC). The Wiegand signal schemes are shown in Table 2 and Figure 9.

Wiegand	D0	D1
nothing	1	1
zeroes	0	1
ones	1	0

Table 2: The Wiegand Physical Layer

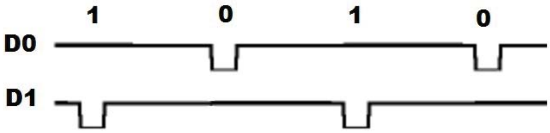
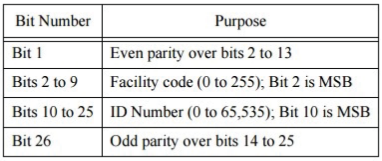


Figure 9: Schedule over Wiegand signals.

Each card has a programmed serial number [43, p. 2]. For 26-bit cards that is the 16-bit ID numbers (0 to 65,535). There is a risk that two different companies share the same card number. To reduce this risk, facility codes are used. The facility code is another 8-bit number (0-255). For example if a company needs 500 card numbers the card numbers could look like follows:

230-0001...230-0500 (Company A)
180-0001...180-0500 (Company B)

This means that the two different companies can use the same ID number with different facility numbers and will not be able to access each others systems. There are also 32-bit and 37-bit card formats to decrease this possibility. For validation the card needs to be granted by the access control unit with both the facility code and ID number. Both the parity bits are used for error detection. The first parity bit is an even parity bit for the first part of the message (bits 2-13). The last bit is an odd parity bit for the second part of the message (bits 14-25). If the error detection and the parity bits are ok then everything is fine. If not, the message has to be sent all over again. Wiegand uses simplex communication.



Bit Number	Purpose
Bit 1	Even parity over bits 2 to 13
Bits 2 to 9	Facility code (0 to 255); Bit 2 is MSB
Bits 10 to 25	ID Number (0 to 65,535); Bit 10 is MSB
Bit 26	Odd parity over bits 14 to 25

Figure 10: A standard Wiegand 26-bit card number format [80].

Pros:
The idea behind Wiegand is very simple. That makes Wiegand easy to implement and use when sending data between two units. When no encryption is needed or low-level of security is applied, Wiegand has the advantage of being a simple interface.
Cons:
One of the major disadvantages with Wiegand is that its wired data transfer is easy to monitor. If the physical connection is accessible it is not hard to implement a sniffer using a simple micro controller (for example an Arduino or a Raspberry Pi). Doing so the data that is sent could be stored in a memory card or transmitted further to the eavesdropper. If the data sent from the card reader is not encrypted, for example sent in plain text, the information is easily reachable for anyone that listens. If the data is encrypted with low-level security there is a possibility that the sniffer might use a given hash table and decrypt the stored information. Wiegand also only supports a distance of up to 150 meters between nodes [58, p. 1].

2.3.2 Clock and Data

According to DSX Access System the clock and data reader are typically used for magnetic stripe cards [32]. The physical layer consists of two wires, the ”clock/strobe” wire transmits the clock signal and ”Data” wire transmits the binary information. The binary information is sent as zeros (0 VDC) or ones (5 VDC) over the ”Data” wire. The ”clock/strobe” line is set to one (5 VDC) each time a bit of data is transmitted on the ”Data” wire, this enables the sampling of the binary information. Clock and data uses simplex. The scheme for the Clock and Data system is shown in Figure 11.

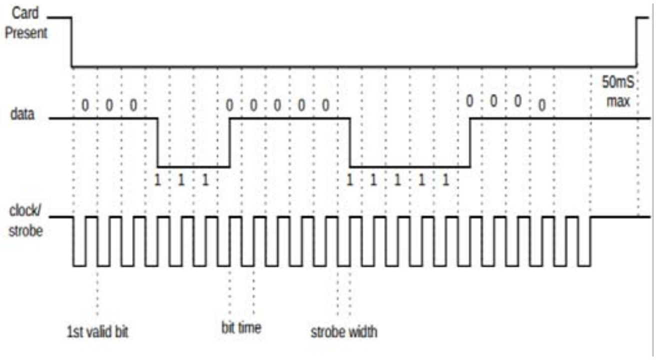


Figure 11: Clock and Data Interface Pulses [42, p. 2].

2.3.3 RS485

Electronics Industry Association (EIA) developed the network standard RS232 for serial communication back in 1962 and many revisions followed. The RS232 was well used in personal computers to connect devices such as printers, modems, mice, etc. However, it suffered many problems with low transmission speed, large voltage swings, and large standard connectors. One of the main problems with the RS232 was that it lacked immunity to signal noise [19]. After a series of network standards (such as RS423 and RS422 in the early 90's) RS485 was approved in 1998. The RS485 standard was a major step forward and offered improvements such as faster transmission speeds and shielded twisted pair wires. Also, with the RS485 up to 32 devices could be connected to the network on the same bus, also called Multidrop Bus (MDB). A MDB enables the devices to send data at different time frames [55, p. 3]. The advantage of MDB's are simplicity, extensibility, and that the devices can send information at any point. This was a step up from the RS422 and RS423 which were only capable of 10 unit-loads on the same line. With some modifications by using a high resistance input that number could be raised to 256 for RS485. A standard RS485 driver handles 32 unit-loads that could consist of 256 one-eights unit-load devices [40]. The RS485 is still widely used today within smaller networks and Electronic Access Control (EAC) systems. RS485 uses half duplex with a pair wire but can also be made full duplex using four wires. Full duplex mode is then limited to Master-Slave situations [4]. In other words, the Master requests information from the slave units and the slave units cannot communicate with each other. The network layer of the RS485 is structured as a daisy chain when the total devices in the network are at most 32. The daisy chain topology is shown in Figure 12.

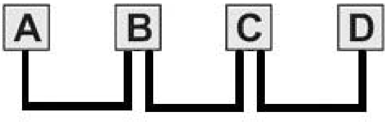


Figure 12: Daisy Chain topology.

If the number of devices connected with RS485 exceed 32, a router device is used to connect multiple network segments with each other. A router device also enables the connectivity to other networks such as TCP/IP over Ethernet with a RS485 network, as shown in Figure 13.

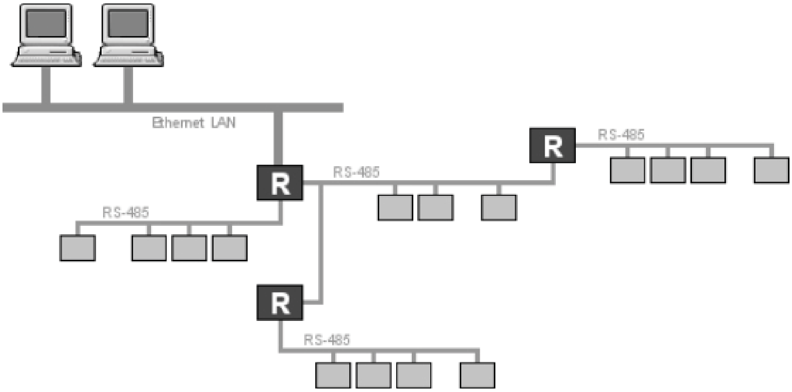


Figure 13: Router connectivity with RS485 networks [55].

Pros:
Some of the advantages with the RS485 standard is that it can be used for long distances, up to 1200m. The shielded twisted pair wires also reduce the signal disturbance significantly which results in more accurate voltage levels. The RS485 can also handle more nodes, up to 32 to be precise (and up to 256 with high resistance inputs). The response time between the reader and the software is also short when less than 32 nodes are connected to the same line [36].
Cons:
RS485 is not well-suited for larger data transfers on long distances. Between 12m and 1200m the speed is about 100 kbps (for data transfers shorter than 12m the speed is about 35 Mbps) [40]. It is not very stable at this speed due to large data losses so the speed has to be reduced to about 56 kbps to get a reliable

connection. At this speed large data transfers will take long time[1]. If more then 32 nodes are used in the network, it will need galvanic isolation. Galvanic isolation is the process of isolating certain parts of the circuit by using magnetic or optical means to prevent ground-loops (unwanted currents in the circuit) [61]. Without galvanic isolation the network will be large enough that electrical noise becomes a factor which will make communication unreliable [56]. Since the RS485 has a MDB structure an idle state may occur when no devices are transmitting, which leaves the transmission line vulnerable to electrical noise. To solve this the controllers (card readers, control units, etc.) should each have a fail-safe circuit which in case of idle state pushes the line to a known state [55, p. 3].

2.3.4 Open Supervised Device Protocol

According to the Security Industry Association (SIA), Open Supervised Device Protocol (OSDP) enables communication between a control unit and peripheral devices (for example card readers) [12, p. 1]. OSDP was originally developed by HID Global and Mercury Security. According to Tony Diodato (CTO at Cypress Integration Solutions) and Joe Gittens (Director of standards at SIA) the Security Industry Association has been assigned ownership of the OSDP specification between a control unit and a card reader [31].

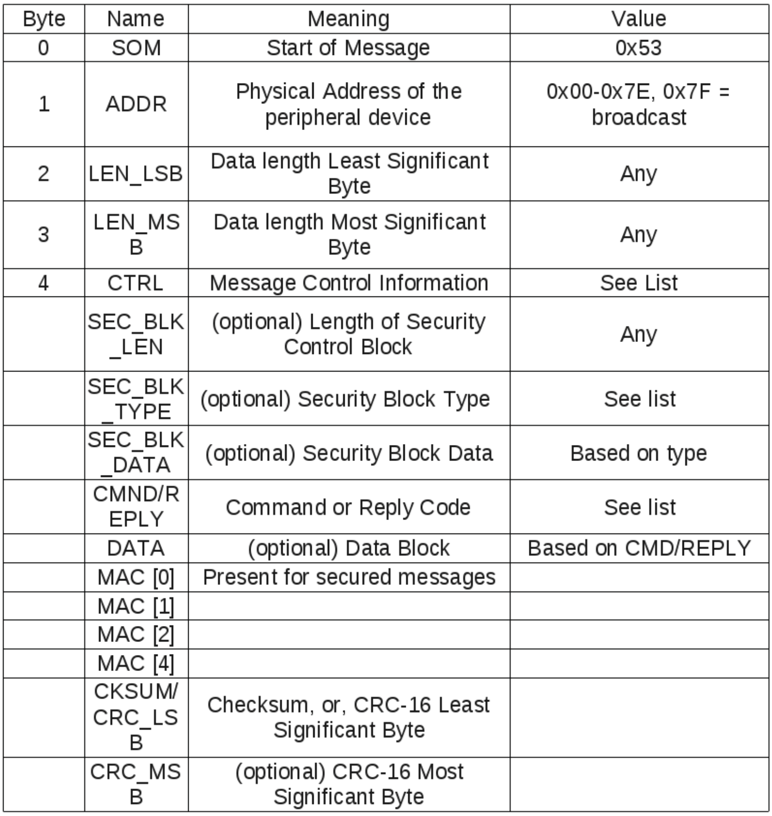
OSDP Transparent Mode is a patented technology by Assa Abloy that gives the alternative to pass messages directly between a smart card and authentication software, thus becoming a tunneling protocol [12, p. 1].

According to the United States Department of Defense OSDP is currently not prescribed for Federal Information Processing Standard Publication 201 (FIPS 201) or Federal Identity Credential and Access Management (FICAM) compliance, thus the majority of the vendors have opted to extend the life of the Wiegand interface instead of implementing OSDP [48, p. 20].

OSDP enables the option to be implemented in both full- and half duplex media with gateway or protocols translators [31]. Its standard implementation is, however, on a multi-dropped RS485 hardware bus with the possibility of connecting 128 readers to the control unit, with a distance up to 1220 meters [3]. Messages between the control unit and its peripheral devices have the structure shown in Figure 14.

---

[1] Everyone that had a 56.6 kbps modem back in the 90's will probably agree.



Byte	Name	Meaning	Value
0	SOM	Start of Message	0x53
1	ADDR	Physical Address of the peripheral device	0x00-0x7E, 0x7F = broadcast
2	LEN_LSB	Data length Least Significant Byte	Any
3	LEN_MSB	Data length Most Significant Byte	Any
4	CTRL	Message Control Information	See List
	SEC_BLK_LEN	(optional) Length of Security Control Block	Any
	SEC_BLK_TYPE	(optional) Security Block Type	See list
	SEC_BLK_DATA	(optional) Security Block Data	Based on type
	CMND/REPLY	Command or Reply Code	See list
	DATA	(optional) Data Block	Based on CMD/REPLY
	MAC [0]	Present for secured messages	
	MAC [1]		
	MAC [2]		
	MAC [4]		
	CKSUM/CRC_LSB	Checksum, or, CRC-16 Least Significant Byte	
	CRC_MSB	(optional) CRC-16 Most Significant Byte	

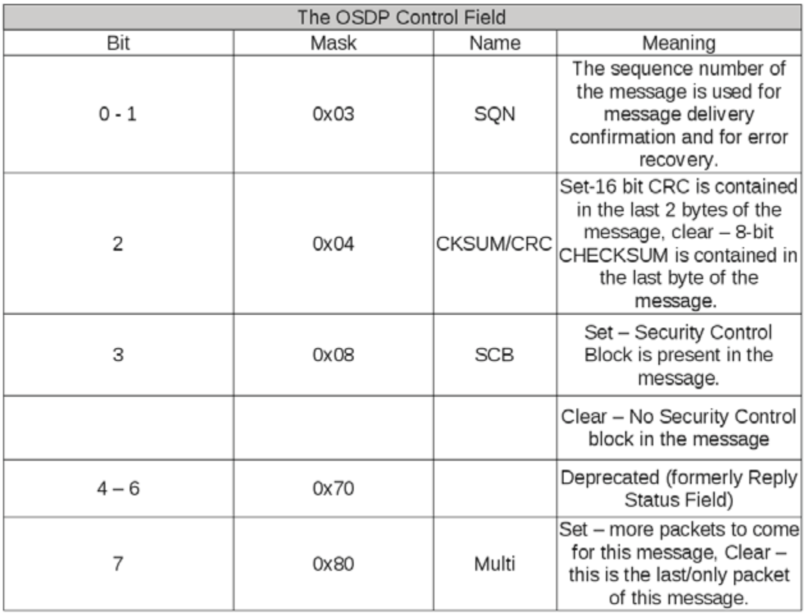
Figure 14: The OSDP Packet Format.

Start of Message:
Start of message signals the beginning of each message header, which is used for synchronization [11, p. 2].

Address:
The address character consists of 8 bits, 7 Least Significant Bit (LSB) that represents the peripheral device address to which the message is sent. The Most Significant Bit (MSB) represents a Broadcast address, when this bit is set all peripheral devices will respond.

Length:
The length field contains the value of the total message length.

Control:
The different parts of the control field are explained in Figure 15.



The OSDP Control Field

Bit	Mask	Name	Meaning
0 - 1	0x03	SQN	The sequence number of the message is used for message delivery confirmation and for error recovery.
2	0x04	CKSUM/CRC	Set-16 bit CRC is contained in the last 2 bytes of the message, clear – 8-bit CHECKSUM is contained in the last byte of the message.
3	0x08	SCB	Set – Security Control Block is present in the message.
			Clear – No Security Control block in the message
4 – 6	0x70		Deprecated (formerly Reply Status Field)
7	0x80	Multi	Set – more packets to come for this message, Clear – this is the last/only packet of this message.

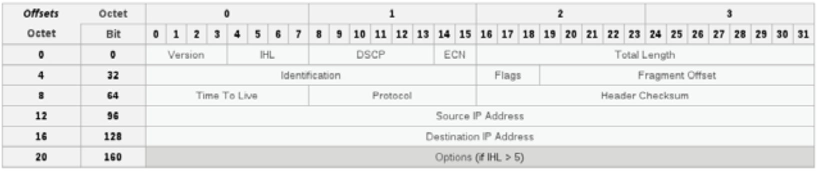
Figure 15: The OSDP Control Field

Security Block:
The Security block (which is optional) enables an implementation of data security in the OSDP framework [11, p. 4].
CMND/Reply - Command/Reply Code:
A command or reply code is generated in this field that states the purpose and meaning of the message [11, p. 5].

Pros:
If half- or full duplex communication is enabled it only takes a small amount of time for the control unit to detect if a peripheral device fails or is tampered with [48, p. 7]. The standard OSDP implementation on a multi-dropped RS485 with daisy-chain has a smaller cable cost then a Wiegand implementation [48, p. 7]. OSDP has support for AES-128 encryption between the control unit and its peripheral devices. The control unit will generate a unique symmetric key for each of its peripheral device using the AES-128 encryption scheme [48, p. 18]. According to the United States Department of Defense, OSDP is set to be implemented at a higher rate as all Federal agencies and departments are under pressure to adopt more secure transport protocols [48, p. 20].
Cons:
The first version of OSDP does not support Secure Channel or Transparent Mode. These attributes were added with version 2 of OSDP. OSDP currently only supports AES-128 (symmetric encryption), thus none of the asymmetric encryption schemes are supported [48, p. 18].

2.3.5 TCP/IP

Another way of communicating that is becoming more interesting to use in EAC is the TCP/IP method. Due to the disadvantages of RS485, manufacturers started to develop new controllers and readers to support TCP/IP connection [36]. Some manufacturers also modify their products by integrating a TCP/IP module. Many application protocols such as the famous File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Secure Socket Layer (SSL) (and many more) use TCP/IP for communication [63]. Unlike RS485, which can only handle 32 drivers on the same line (up to 256 in some cases), TCP/IP enables IP addresses. IPv4 addresses contains a 4 set of 3-digit numbers[2] which results in about 4.3 billion addresses in total[3]. The IPv4 header is shown in figure 16 and contains 20 bytes. The whole package length when maximum data is sent is 65,535 bytes (including the header). For example if no data is sent the package will have a minimum length of 20 bytes, which only includes the header.



<table>
<tr><th><i>Offsets</i></th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td><b>0</b></td><td><b>0</b></td><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><td><b>4</b></td><td><b>32</b></td><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td><b>8</b></td><td><b>64</b></td><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td><b>12</b></td><td><b>96</b></td><td colspan="32">Source IP Address</td></tr>
<tr><td><b>16</b></td><td><b>128</b></td><td colspan="32">Destination IP Address</td></tr>
<tr><td><b>20</b></td><td><b>160</b></td><td colspan="32">Options (if IHL > 5)</td></tr>
</table>

Figure 16: IPv4 Header Format [93].

A common used cable is the Category 5 (CAT5) which can be run in half-duplex or full duplex mode with a speed of 1000 Mbps (up to 100 m). TCP/IP also enables the potential of Power over Ethernet (PoE) for implementation with the CAT5 (or newer version) cable. PoE can deliver up to 25 watts of power in addition to the data transferred with its newest standard IEEE 802.3at-2009 [10]. The Axis A1001 Network Door Controller has PoE support [25]. Worth mentioning is that with some modifications (without breaking the standard) it is also possible to double that power, up to 51W. One example of this is the Axis Q6032-E PTZ Network Camera which is directly powered with High PoE [24].

Pros:
The benefits of using TCP/IP are many. It is a well known protocol and used in many applications. Compared to RS485 which can only be used in small networks, TCP/IP has a huge advantage using IP addresses. The possibility to use PoE can also be of great benefit in many EAC products since it only requires one cable for both power and data communication. Other benefits are: reliable data delivery, low data overhead, platform independence (available for all modern operating systems) and the ability to add networks without interrupting existing services [77].

---

[2] For example 255.255.255.255

[3] Due to the major establishment of computers, smartphones, tablets etc. in the world the IP addresses are just running out. IPv6 is nowadays replacing IPv4 to solve this problem.

Cons:
Currently there are very few known disadvantages with TCP/IP. It may be intricate to set up but with focus on the Axis A1001 Network Door Controller that will not be a problem because it already supports TCP/IP [25]. TCP/IP may also be slow for smaller networks. In smaller networks and intranets running checksums can cause machine lags due to memory demands, and eat up bandwidth [64].

2.4 Smart card protocols

This chapter will go through different smart card protocols that will be evaluated in the thesis.

2.4.1 MIFARE

MIFARE is a trademark for a series of chips owned by NXP Semiconductors. MIFARE is used in contactless smart cards and proximity cards and complies with the ISO/IEC 14443 standard (used in over 80% of all contactless smart cards today [84]). The first MIFARE product was introduced in 1994 and was used in the public transport ticketing system in Seoul, Korea. Since then the usage of MIFARE and such smart card solutions have increased drastically.

All of the MIFARE integrated circuits are operating at the standardized 13.56 MHz frequency range and transfers data with a rate of 848 Kbit/s. Their first MIFARE Classic offers only a 48-bit key size with its proprietary Crypto-1 cipher mechanism, developed by NXP. MIFARE Classic is used widely in public transport ticketing[4] and access control systems but is now considered broken [37, p. 2]. Thus MIFARE recommend using their higher security product families such as MIFARE DESFire [85].

2.4.2 MIFARE DESFire

DESFire was introduced in 2002 and is a variant of MIFARE. The first three letters refers to the 3DES symmetric cryptographic scheme while the rest are "Fast, Innovative, Reliable and Enhanced". It contains more features than MIFARE Classic and is considered a more secure alternative solution. The DESFire EV1 (announced in 2006) supports a 128-bit Advanced Encryption Standard (AES) key or 56-bit, 112-bit or 168-bit 3DES encryption [86]. EV1 is also compatible with NFC which enables integration with mobile devices. According to David Oswald and Christof Paar (a professor and a doctoral student working with IT-Security at Ruhr-University in Bochurm, Germany) the DESFire with up to 112-bit 3DES algorithm can be broken quite easily with Side-Channel Attacks and is considered a non-secure solution [74]. This does not affect DESFire EV1 though, because the newer version has enhanced security thus they received the Evaluation Assurance Level (EAL) 4+ grade. Therefore it is recommended to use DESFire EV1 with AES-128 key encryption.

The DESFire EV2 (announced in 2013) is the latest evolution of DESFire and

---

[4] For example in the JoJo cards used by Skånetrafiken (a Swedish public transport company).

contains the same cryptographic properties as EV1 but includes more overall features. For example it can hold unlimited different applications and the only limitation is the memory size of the card. Worth mentioning is that EV2 is backward compatible with previous evolutions.

2.4.3 iClass

HID based the iClass card on a PicoPass card, configuring the original design of the PicoPass card so that its setting can not be modified [37, p. 4]. iClass contains a number of design and implementation faults, such as the usage of weak keys, XOR key update weaknesses, weak key diversification and privilege escalation weakness [37, p. 11-12]. All of the mentioned weaknesses were taken advantage of in an key recovery attack on iClass [37, p. 3]. The key recovery attack was initiated by eavesdropping on a successful authentication session. Using the sniffed information the scientist carrying out the attack ran 2^{22} authentication attempts and 2^{19} key updates while using a radio frequency identification tool, this enabled the scientist to gain 24 bits of the card key. The remaining 40 bits were recovered using 2^{40} off-line MAC computations. The scientist had now recovered the card key, acquiring the master key from the card key was then a matter of penetrating a single Data Encryption Standard (DES) algorithm.

2.4.4 iClass Elite

iClass Elite differentiate itself from iClass by enabling its users to have a unique master key for each system [37, p. 13]. An attribute that is achieved by adding an extra step to the diversification algorithm used by regular iClass. iClass Elite diversification algorithm has been reversed engineered by a group of scientist that decided to publish it in full detail [37]. iClass Elite contains a number of design and implementation faults, such as redundant key diversification and weak key-byte selection [37, p. 14-15]. The mentioned weaknesses were taken advantage of in a key recovery attack on iClass Elite [37, p. 3]. The scientist conducting the attack ran 15 authentication attempts on a iClass card reader. The attack was then followed up with an offline computation of 2^{25} DES encryptions resulting in the scientist gaining the master key.

2.5 Government standards and protocols

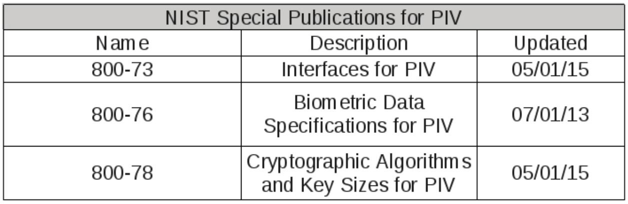
There are wide quality and security variations used to gain access to secure facilities. To eliminate these variations and reduce the potential threat of terrorist attacks, the White House released Homeland Security Presidential Directive 12 (HSPD-12) signed by George W. Bush on August 27, 2004 [92]. The U.S. policy is to increase Government efficiency, reduce identity fraud, enhance security, and protect personal privacy by establishing a mandatory Government-wide standard for secure and reliable forms of identification. These identifications are issued by the federal Government to its employees and contractors. In response to HSPD-12 the National Institute of Standards and Technology (NIST) Computer Security Division initiated a program for improving identification and authentication of federal employees and contractors [70]. FIPS 201 was developed to satisfy the HSPD-12 directive and was approved on February 25, 2005.

More details about FIPS 201 will be described in Chapter 2.5.1. There are a large number of NIST Special Publications designed specifically to enhance security in various systems and fields. A full list of those can be found on the NIST website [72].

The release of FIPS 201 marked the beginning of a new development and validation phase for both private sector and federal departments and agencies. By 2009, over 300 products had been developed, validated and released to the market to support the Personal Identity Verification (PIV) standards and their infrastructure.

2.5.1 FIPS-201

Federal Information Processing Standard Publication 201, FIPS 201, incorporates three of the NIST Special Publications. In Figure 17 the three publications are shown which are required for a PIV system [70]. In August 2013, some high-level changes were made and a revision of the original FIPS 201 was released, called FIPS 201-2 [23]. FIPS 201-2 specify the minimum requirements for a federal PIV system that meets the control and security objectives of HSPD-12, including identity proofing, issuance, and registration. It also specifies the physical card characteristics, storage media, and data elements that make up identity credentials. This standard does not include specifications of access control policies or requirements for federal departments and agencies.



NIST Special Publications for PIV		
Name	Description	Updated
800-73	Interfaces for PIV	05/01/15
800-76	Biometric Data Specifications for PIV	07/01/13
800-78	Cryptographic Algorithms and Key Sizes for PIV	05/01/15

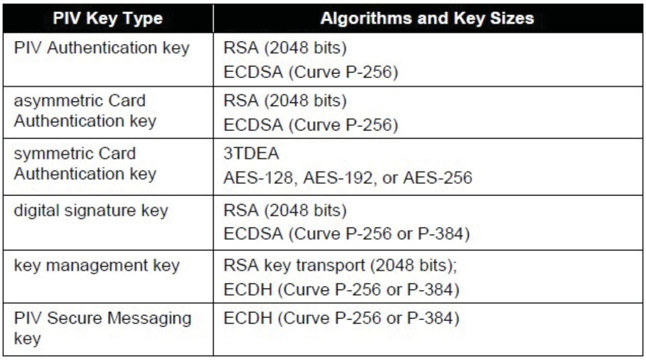
Figure 17: Publications required for a PIV system.

How storing and receiving identity credentials from a smart card works is specified in [SP 800-73-4], "Interfaces for Personal Identity Verification" [71]. The requirements for cryptographic algorithms are specified in [SP 800-78-4], "Cryptographic Algorithms and Key Sizes for PIV" [27].

According to [SP 800-73-4] and [SP 800-78-4] the PIV card must store private keys and corresponding public key certificates and perform cryptographic operations using the asymmetric private keys [23].

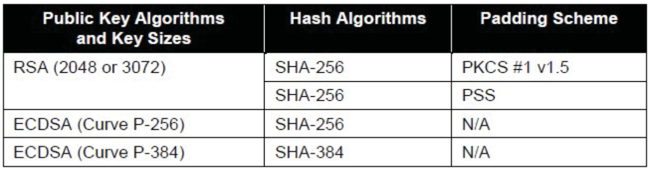
In Chapter 3 ("On Card Cryptographic Requirements") of the [SP 800-78-4], it is said that all of the objects stored on the card may be divided into three classes. These are: *PIV Cryptographic keys*, *Signed Authentication Information Stored on the PIV Card*, and *Message Digests of Information Stored on the PIV Card*.

The minimum requirement for the PIV cards is that it must store two asymmetric private keys and the corresponding public key certificates. They are called PIV *Authentication Key* and the *asymmetric Card Authentication Key* (the top two Key Types in Figure 18). The PIV cards must also store a digital signature key, a key management key and their corresponding public key certificate for cardholders that have government-issued email accounts. The remaining two keys are optional to use.



PIV Key Type	Algorithms and Key Sizes
PIV Authentication key	RSA (2048 bits) ECDSA (Curve P-256)
asymmetric Card Authentication key	RSA (2048 bits) ECDSA (Curve P-256)
symmetric Card Authentication key	3TDEA AES-128, AES-192, or AES-256
digital signature key	RSA (2048 bits) ECDSA (Curve P-256 or P-384)
key management key	RSA key transport (2048 bits); ECDH (Curve P-256 or P-384)
PIV Secure Messaging key	ECDH (Curve P-256 or P-384)

Figure 18: Algorithms and Key Size requirements for PIV Key types. Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie Hellman (ECDH) are two elliptic curve algorithms (Digital Signature Algorithm and Diffie-Hellman) [27, p. 6].



Public Key Algorithms and Key Sizes	Hash Algorithms	Padding Scheme
RSA (2048 or 3072)	SHA-256	PKCS #1 v1.5
	SHA-256	PSS
ECDSA (Curve P-256)	SHA-256	N/A
ECDSA (Curve P-384)	SHA-384	N/A

Figure 19: Signature Algorithm and Key Size Requirement for PIV Information. PKCS stands for Public-Key Cryptographic Standards and PSS stands for Probabilistic Signature Scheme. [27, p. 7]

FIPS 201-2 requires digital signatures to protect integrity and authenticity of stored information on the PIV card (the objects that require digital signatures are specified in [SP 800-78-4]). Figure 19 describes what public key algorithms, key sizes, hash algorithms and padding schemes are required to generate digital signatures for digitally signed information stored on the PIV cards.

2.5.2 PIV

PIV is a credential and standard background investigation process required by the presidential directive HSPD-12 [67]. The goal of the PIV program was to meet new security standards effectively and cost efficient. A PIV card is a United States Federal smart card used by the cardholders to grant access to Federal facilities and information systems with the appropriate level of security. The criteria for the PIV card was established with FIPS 201. Since 2005 the U.S government has delivered over 5 million PIV cards to Federal employees and contractors. The PIV cards are following FIPS 201 and must follow [SP 800-73] [8]. The credential numbers on the card are in Federal Agency Smart Credential Number (FASC-N) format, which requires a Federal agency code. This also implies a national agency background check with investigation. This is the highest security level of the different PIV standards.

”The PIV card is an identity card that conforms fully to Federal PIV standards. Only cards issued by Federal entities can fully conform. Federal standards ensure that PIV cards are interoperable with and trusted by all Federal Government relying parties.” [7]

2.5.3 PIV-I

Within the private sector there exist a large demand for more secure systems and standards, leading to many companies looking in the direction of PIV because of its success. Non-Federal issuers of identity cards wanted to issue identity cards that can be trusted by the Federal government and also interoperable with Federal government PIV systems [7]. In May 2009 the Federal CIO Council published a guidance document called *Personal Identity Verification Interoperability for Non-Federal Issuers*. This document specifies the minimum requirements for non-Federal issued identity cards Personal Identity Verification Interoperability (PIV-I) cards that can technically interoperate with Federal government PIV systems and that are trusted by the Federal government relying parties. PIV-I follows FIPS 201 and must follow [SP 800-73] [8]. These are the two major differences compared to PIV. PIV-I also uses the credential number in Universal Unique Identifier (UUID) format instead of FASC-N, this means that no Federal agency code is required.

”The PIV-I card is an identity card that meets the PIV technical specifications, works with PIV infrastructure elements, such as card readers, and is issued in a manner that allows Federal Government relying parties to trust the card.” [7]

The use of PIV-I supports the objectives of ICAM (described in Chapter 2.5.6) as it allows an agency to have strong security in interaction with external business partners. The advantage of PIV-I is that is does not require any background checks, it follows FIPS 201 and still keeps a high level of security. Another type is the Personal Identity Verification Compatible (PIV-C) card. A state or a local government can choose to implement PIV-I or a PIV-C card depending on the appropriate security level. The PIV-I card builds on the PIV-C card but is issued in a manner consistent with FIPS 201 policies and processes. Thus the PIV-I card can be trusted by both the state, the local government and the federal government.

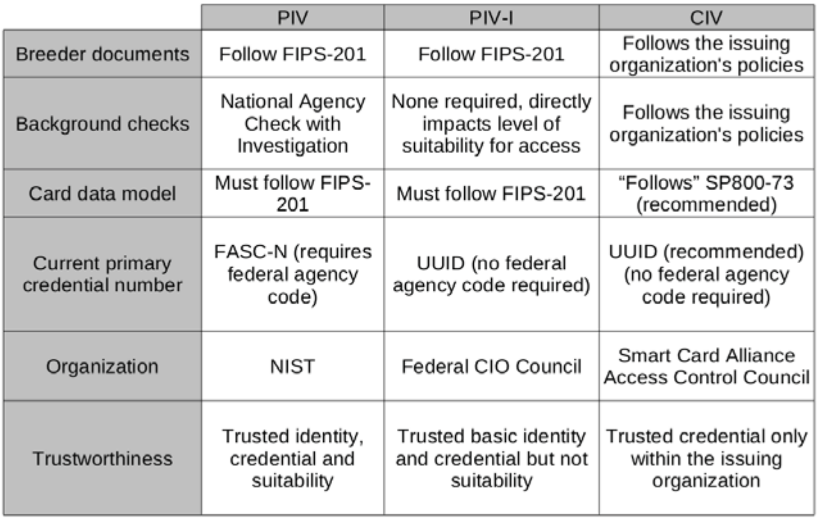
”The PIV-C card is an identity card that meets the PIV technical specifications:

the card can work with PIV infrastructure elements, such as card readers, but the card itself has not necessarily been issued in a manner that assures it is trustworthy by Federal Government relying parties.” [7]

A PIV-C card would not be trusted by the Federal government relying parties. It would still be technically compatible with PIV infrastructure elements such as card readers. To support the PIV infrastructure the authorities issuing these cards can then take advantage of the increasing number of products that are available. PIV-C does not have as high security level as the other two PIV standards (PIV and PIV-I) in such ways that they are not trusted by Federal government or its relying parties.

2.5.4 CIV

When PIV-C was first released it gained a bad reputation [81]. A result of ’Compatibility’ being too close to ’Interoperability’ and thus a new term was requested. Smart Card Alliance (SCA) released a White Paper describing Commercial Identification Verification (CIV) (Commercial Identity Verification) in October 2011 in order to replace PIV-C. CIV does not have the same security level as PIV and PIV-I but is focused on the commercial use where not as high security level is required. Note that this means that the issuers can still implement very high security protocols and standards in the cards, but the main difference is that these cards are not trusted by the Federal government and their relying parties. Figure 20 shows a comparison between the different versions of PIV cards.



	PIV	PIV-I	CIV
Breeder documents	Follow FIPS-201	Follow FIPS-201	Follows the issuing organization's policies
Background checks	National Agency Check with Investigation	None required, directly impacts level of suitability for access	Follows the issuing organization's policies
Card data model	Must follow FIPS-201	Must follow FIPS-201	“Follows” SP800-73 (recommended)
Current primary credential number	FASC-N (requires federal agency code)	UUID (no federal agency code required)	UUID (recommended) (no federal agency code required)
Organization	NIST	Federal CIO Council	Smart Card Alliance Access Control Council
Trustworthiness	Trusted identity, credential and suitability	Trusted basic identity and credential but not suitability	Trusted credential only within the issuing organization

Figure 20: This table shows the most important factors and requirements in a comparison between PIV, PIV-I and CIV.

According to SCA the CIV credentials are technically compatible with the PIV-

I specifications. The difference is that a CIV credential issuer does not need to follow the strict policy framework of PIV and PIV-I when it comes to both issuance and use. This allows a certain freedom to the companies to deploy the standardized technologies in such way that are suitable to their own environments. One major difference is that CIV does not need to follow FIPS 201 and follows the issuing organization's policies instead [8]. CIV is only recommended to follow [SP 800-73] in its data card model. It is also recommended to use UUID as credential numbers, but it is not required. SCA Physical Access Council defined CIV credential and established the availability of a PIV-level credential for commercial use that replaced the PIV-C concept [44]. This way companies can still achieve a high level of security for their products without the approval and background checks from the government. Note that the trustworthiness, or the trusted credential are only within the issuing organization.

Oak Ridge National Labs (ORNL), a technology research facility for the U.S. Department of Energy deployed a mix of PIV and CIV cards in their Tennessee facilities in August 2014 [65]. The new smart card credentials were used both for physical and logical access. Oak Ridge ordered PIV cards to the employees that needed to travel and use their credentials at other facilities. The employees that only worked at a specific site were given CIV cards. The CIV cards are less expensive so if thousands of employees are supposed to get credentials, this is a more cost efficient way to go while keeping a high level of security. FIPS 201 and PIV are not mandated for Oak Ridge but the work they are doing in the Federal space require them. This is a good example of how CIV could be used together with PIV systems and how flexible and compatible this credential standard is.

The company issuing PIV cards to ORNL is Gemalto [78]. One of their smart card families, the one used in ORNL, is called IDPrime. The IDPrime card is a PIV credential standardized card which can be used for private sector and government organizations [39]. To meet the ORNL's requirements Gemalto cooperated with Charismatics and Quantum Secure.

2.5.5 Derived PIV

FIPS 201 enables PIV credentials to be stored on a smartphone if the keys are stored within a HSM thus enabling the smartphone to replace a PIV Card, this is called Derived PIV [35, p. 1]. The Derived PIV credential consists of a X.509 public key certificate that is issued in accordance with PIV standards [35, p. 3]. The approved cryptographic algorithms for Derived PIV are the same as for PIV, thus these are shown in Figure 18 [35, p. 10]. The HSM used with Derived PIV may either be embedded inside the smartphone or a removable item. The removable items that PIV permits as HSM implementations are Secure Digital (SD) cards, Universal Integrated Circuit Card (UICC), and Universal Serial Bus (USB) devices [35, p. 11].

Implementing Derived PIV on the smartphone embedded HSM is either done fully in hardware or software, however, embedded software-based security modules are at a higher risk to be stolen or compromised thus their level of assurance is lower than hardware implemented HSM [35, p. 12]. NIST suggests protecting the derived credentials stored within a software-based HSM by encrypting them and using a password with a 6-digit PIN [35, p. 13]. A 6-digit password corresponds to 20 bits of entropy leaving it vulnerable to an offline brute force attack, thus implementing Derived PIV on a software-based HSM is not recommended. The guidelines NIST have for the embedded hardware-based HSM solutions are deeply lacking, and have come under the scrutiny of different companies and Federal agencies trying to implement the solutions [28]. Apple refers to the guidance by NIST in this area as being extremely weak. National Security Agency (NSA) referred to the guidance by NIST as being wrongly focused on removable HSM as the market has a lack of interest in this. Thus NSA projects that most agencies are going to try to implement embedded solutions and the lack of guidance by NIST in this area is of major concern. Tech Companies, for example Microsoft, made the suggestion of using a Trusted Platform Module, a Trusted Execution Environment or a Secure Element as an embedded hardware-based HSM. All of these suggestions are, however, omitted from the latest NIST guidance. The latest NIST guidance details what NIST refers to as a hybrid approach were the key is stored in hardware and an embedded software-based HSM uses the key during the authentication process. The hardware that can be used is, however, undefined, and there is no real guidance on how this hybrid approach can be implemented [35, p. 12]. The latest NIST guidance allows Derived and original PIV credentials to be distributed by two separate organizations, which in turn creates major security concerns [28]. The two separate organizations need to be fully synchronized regarding revocation of compromised PIV credentials. If one of the PIV credentials is compromised both need to be revoked, something that can be challenging if the organizations are not fully synchronized.

NSA states that UICC shall not be used as a removable HSM, as the UICC will never be explicitly under the control of the issuing agency. Thus the user can make undetectable modifications to the UICC and thus it should be removed from the NIST guidance. The latest NIST guidance has, however, not removed UICC as an acceptable solution.

2.5.6 FICAM

The FICAM initiative seeks to unite and strengthen the identity, credential and access management activities used in the United States Government [34]. The "FICAM Roadmap and Implementation Guidance" contains the architecture of FICAM and is used to implement ICAM in the United States Government. FICAM stands for Federal ICAM.

ICAM is used to bind identities to credentials within a US agency, thus cutting across all offices, programs and systems within the agency [6, p. 8]. Figure 21 shows how different parts of an agency interconnects using ICAM, as divided into four different parts. "Identity Management" is the part of ICAM that defines the process of giving specific attributes to a digital identity that belongs to a specific individual [6, p. 9]. "Credential Management" defines the process of binding a persons identity to a token so that person can gain access to his or hers specific attributes. "Access Management" defines rules and regulations for physical- and logical access to different parts of the agency. Access to these areas depends on the persons attributes.

The ICAM Services Framework shown in Figure 22 defines the different parts of the ICAM segment architecture [6, p. 14].

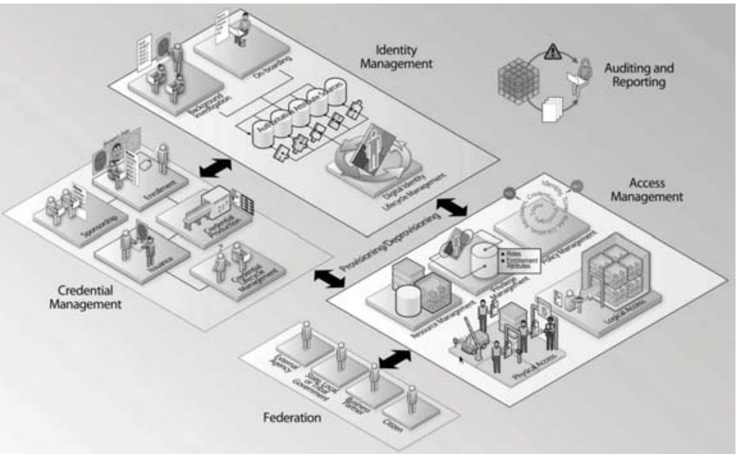
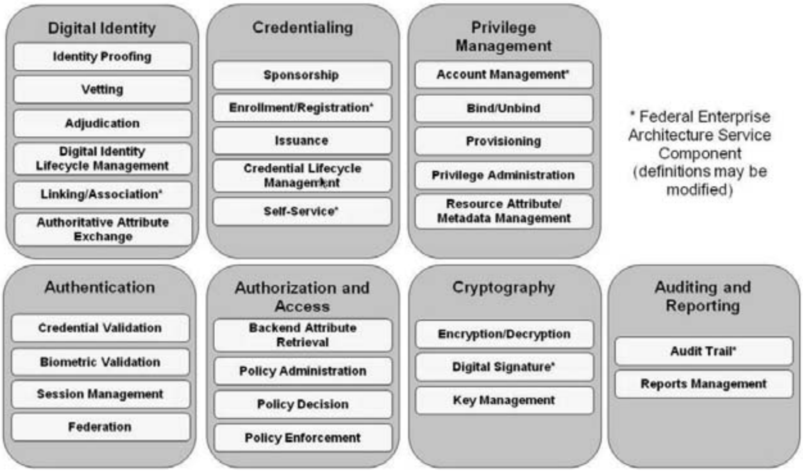


Figure 21: ICAM Conceptual Diagram [6, p. 8].



Digital Identity
- Identity Proofing
- Vetting
- Adjudication
- Digital Identity Lifecycle Management
- Linking/Association*
- Authoritative Attribute Exchange

Credentialing
- Sponsorship
- Enrollment/Registration*
- Issuance
- Credential Lifecycle Management
- Self-Service*

Privilege Management
- Account Management*
- Bind/Unbind
- Provisioning
- Privilege Administration
- Resource Attribute/ Metadata Management

* Federal Enterprise Architecture Service Component (definitions may be modified)

Authentication
- Credential Validation
- Biometric Validation
- Session Management
- Federation

Authorization and Access
- Backend Attribute Retrieval
- Policy Administration
- Policy Decision
- Policy Enforcement

Cryptography
- Encryption/Decryption
- Digital Signature*
- Key Management

Auditing and Reporting
- Audit Trail*
- Reports Management

Figure 22: ICAM Services Framework [6, p. 14].

2.5.7 PLAID

Protocol for Lightweight Authentication of Identity (PLAID) is a protocol that uses asymmetric Rivest Shamir Adleman (RSA) and symmetric (AES) cryptography to protect the communication between an ICC and terminal-devices

[68, p. 3]. The commonwealth of Australia owns the property rights and source code of PLAID [68, p. 5]. PLAID does not deal with management of keys in the cryptographic system. Key management is instead left to be determined by the implementers of PLAID [68, p. 6]. The PLAID protocol is described in Figure 23. An explanation of the different variables in the protocol is shown in Figure 24.

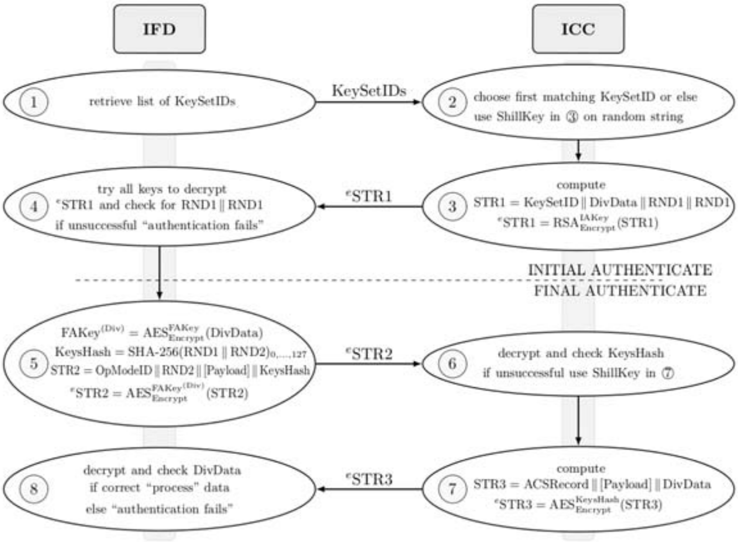
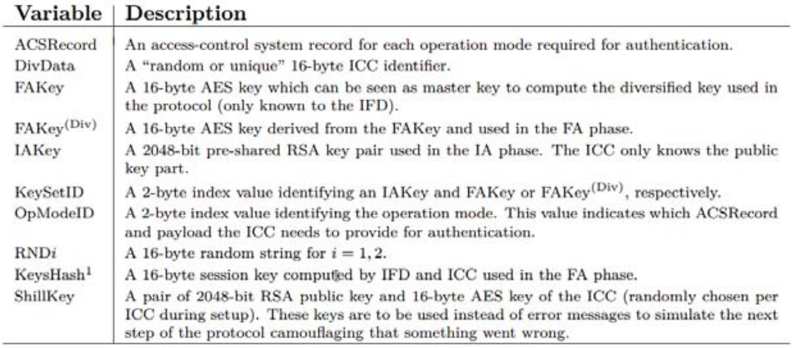

IFD
ICC
① retrieve list of KeySetIDs
KeySetIDs
② choose first matching KeySetID or else use ShillKey in ③ on random string
③ compute $STR1 = KeySetID \| DivData \| RND1 \| RND1$ $^{e}STR1 = RSA^{IAKey}_{Encrypt}(STR1)$
$^{e}STR1$
④ try all keys to decrypt $^{e}STR1$ and check for $RND1 \| RND1$ if unsuccessful "authentication fails"
INITIAL AUTHENTICATE
FINAL AUTHENTICATE
⑤ $FAKey^{(Div)} = AES^{FAKey}_{Encrypt}(DivData)$ $KeysHash = SHA\text{-}256(RND1 \| RND2)_{0,\dots,127}$ $STR2 = OpModeID \| RND2 \| [Payload] \| KeysHash$ $^{e}STR2 = AES^{FAKey^{(Div)}}_{Encrypt}(STR2)$
$^{e}STR2$
⑥ decrypt and check KeysHash if unsuccessful use ShillKey in ⑦
⑦ compute $STR3 = ACSRecord \| [Payload] \| DivData$ $^{e}STR3 = AES^{KeysHash}_{Encrypt}(STR3)$
$^{e}STR3$
⑧ decrypt and check DivData if correct "process" data else "authentication fails"

Figure 23: PLAID protocol overview [29, p. 4].



Variable	Description
ACSRecord	An access-control system record for each operation mode required for authentication.
DivData	A "random or unique" 16-byte ICC identifier.
FAKey	A 16-byte AES key which can be seen as master key to compute the diversified key used in the protocol (only known to the IFD).
$FAKey^{(Div)}$	A 16-byte AES key derived from the FAKey and used in the FA phase.
IAKey	A 2048-bit pre-shared RSA key pair used in the IA phase. The ICC only knows the public key part.
KeySetID	A 2-byte index value identifying an IAKey and FAKey or $FAKey^{(Div)}$, respectively.
OpModeID	A 2-byte index value identifying the operation mode. This value indicates which ACSRecord and payload the ICC needs to provide for authentication.
RNDi	A 16-byte random string for $i = 1, 2$.
KeysHash[1]	A 16-byte session key computed by IFD and ICC used in the FA phase.
ShillKey	A pair of 2048-bit RSA public key and 16-byte AES key of the ICC (randomly chosen per ICC during setup). These keys are to be used instead of error messages to simulate the next step of the protocol camouflaging that something went wrong.

Figure 24: PLAID Variables [29, p. 6].

PLAID and The German Tank Problem:
PLAID uses shill keys. As stated in Figure 24 the shill key is a RSA key that

is supposed to protect the user from ID-Leakage by sending encrypted random data instead of an error messages [29, p. 3]. The main issue with this is that each ICC has a unique shill key and thus a unique RSA cipher text. This in turn enables ID-Leakage.

The German Tank problem has its roots in World War 2 when the allied forces wanted to estimate the number of tanks produced by the Germans on a monthly basis. The Allied Forces solved this by analyzing the serial numbers of downed German tanks, and inventing a mathematical equation that gave the number of tanks based on serial number acquisition [47]. The same equation (Equation 1) can be used to obtain the shill key RSA modulus ”n” that is unique for each ICC [29, p. 9].

$$\hat{N}_j = m_j + \frac{m_j}{k_1} \tag{1}$$

The shill key attack consist of Eve obtaining a k1 large number of Cipher texts from each ICC. Eve then uses the obtained samples to guess which arbitrary Cipher text belongs to a certain ICC by using Equation 1 [29, p. 9]. A simulation of the shill key attack is shown in Figure 25, where k2 is the number of arbitrary cipher texts and the baseline shows the success of purely guessing which card the cipher text belongs to [29, p. 10]. Other design aspects of PLAID also make it vulnerable to both padding oracle attacks and forward secrecy compromisation [29, p. 19].

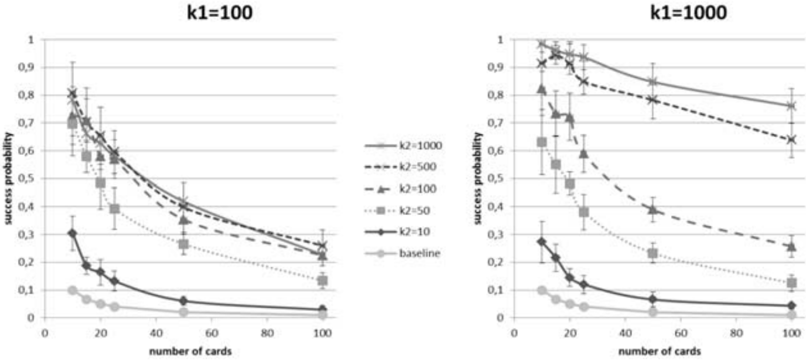


Figure 25: Shill Key Attack with k1=100 samples or k1=1000 samples. The graph shows the success probability of the attack based on the number of cards attacked [29, p. 10].

2.6 Symmetric Cryptographic Schemes

This chapter will go through several symmetric cryptographic schemes that are of interest for the thesis.

2.6.1 Block cipher mode of operation

The block cipher mode of operation contains a symmetric key block cipher algorithm and is used to give communications confidentiality and authentication possibilities [69]. Padding is used so that the number of bits in the plaintext is a positive multiple of the block size [33]. Padding is usually done by appending extra bits to the end of the plaintext. Electronic Codebook (ECB) and CBC are two different modes of operations.

Electronic Codebook:
In ECB each ciphertext is encrypted and decrypted separately using a shared key, this method preserves redundancy hence using ECB is not recommended [53]. The ECB scheme for encryption of plaintext and decryption of ciphertext is shown in Figure 26.

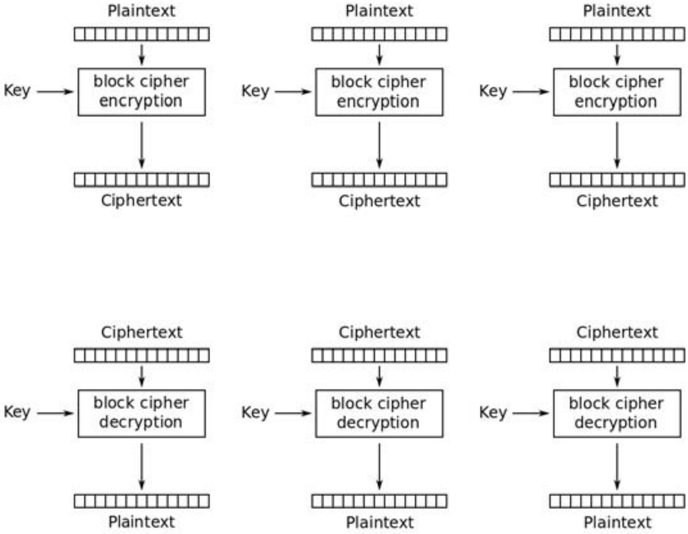


Figure 26: ECB encryption and decryption [2].

Cipher Block Chaining:
In Cipher Block Chaining (CBC) the plaintext is XOR:ed with the previous encrypted ciphertext block, the first block, however, uses an initialization vector to ensure that redundancy is not a factor [53]. The CBC scheme for encryption of plaintext and decryption of ciphertext is shown in Figure 27. [2].

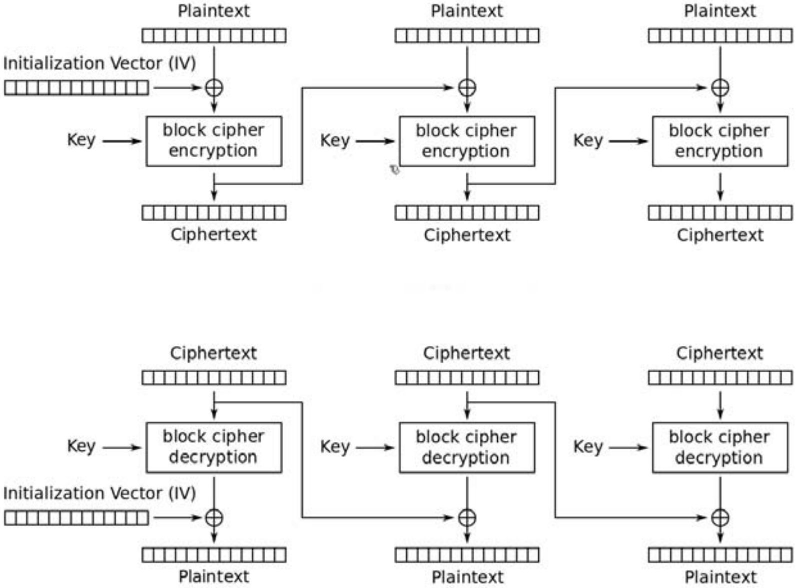


Figure 27: CBC encryption and decryption [2].

2.6.2 DES

DES is a block cipher that was first put into widespread use in 1976 after the National Bureau of Standards (now known as The National Institute of Standards and Technology) adopted the IBM developed cryptographic scheme [73]. The DES had a life span that lasted until the scheme was no longer considered secure as it could be broken by a brute force attack, which happened in 1997. DES is based around the usage of Feistel networks, which gives the ability to build a block cipher from arbitrary functions. More specifically DES uses a 16 round Feistel network. The structure of a one round Feistel network is shown in Figure 28.

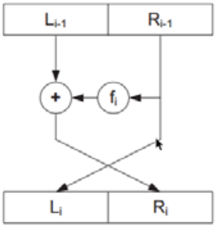


Figure 28: One round Feistel network [14, p. 2].

A round in the Feistel network is structured as in the following steps [14, p. 1]:
1. The input is divided into two parts L_{i-1} and R_{i-1}
2. A round function f_i it then applied so that $f_i(R_{i-1})$.
3. Step 2 is XOR:ed with L_{i-1} yielding $L_{i-1} \oplus f_i(R_{i-1})$.
4. The left and right side switches places thus the output is $L_{i-1} \oplus f_i(R_{i-1})$ and R_{i-1}.

Triple Data Encryption Standard:
An initiative to strengthen DES is using 3DES, which consist of cascading three DES (each with unique keys) [14, p. 11].

2.6.3 AES

AES is a block cipher that was designed by Joan Daemen and Vincent Rijmen in 1997, it was originally named Rijndael as a mix between the two designers last names [90]. Rijndael gained the name AES after winning a cryptography competition held by the National Institute of Standards and Technology that ended in 1999. Today AES is the most widely used symmetric cryptographic scheme in the world. With correct implementation no attacks other than brute-force attacks are known to work on AES. However, because of the large key length of either 128, 192 or 256 bits a brute force attack is not possible within a realistic time frame with modern technology [95, p. 22]. The encryption and decryption of AES is shown in Figure 29.

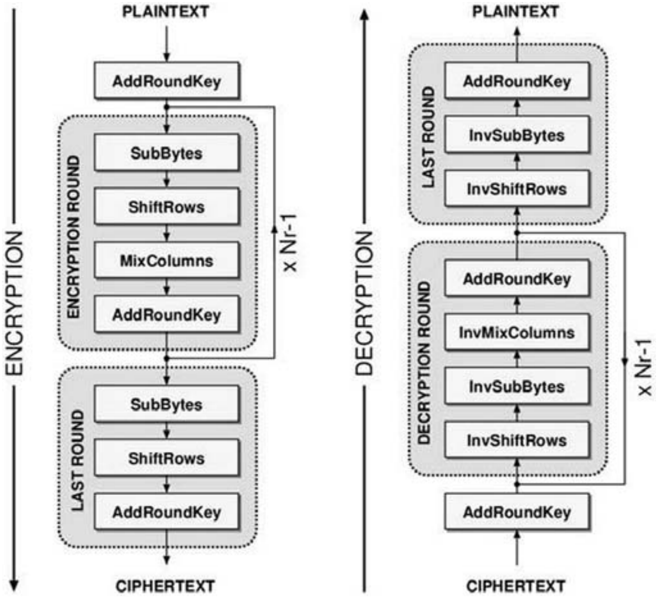


Figure 29: AES encryption and decryption [51].

The number of rounds (Nr) is dependent on the key size. This correlation is shown in Table 3 [51].

Key Size (bits)	Number of Rounds
128	10
192	12
256	14

Table 3: AES key round correlation

The AddRoundKey operation XOR:s a round Key with the current block. The subBytes operation ads nonlinear elements to the scheme, thus confusion is achieved [90]. The ShiftRow operation rearranges the data on a byte level. The MixColumns operation is a matrix operation that rearranges blocks of four bytes. Thus ShiftRow and MixColumns provide diffusion to the AES scheme. The internal structure of the encryption part is shown in Figure 30.

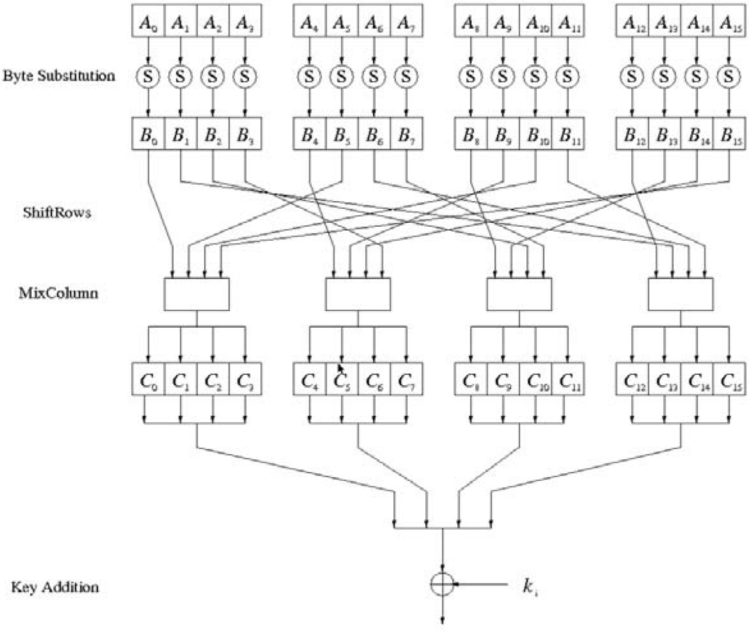

A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}
Byte Substitution
S
B_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15}
ShiftRows
MixColumn
C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14} C_{15}
Key Addition
k_i

Figure 30: AES encryption in detail [95, p. 7].

2.7 Asymmetric Cryptographic Schemes

In general, asymmetric cryptographic schemes consist of a public key and a private key. In some cases or modifications of algorithms they use two of each key. The two parties Alice and Bob share their public keys with each other and they are used for encryption while the private keys are used for decryption.

2.7.1 RSA

Rivest Shamir Adleman (RSA) is an algorithm which was first published in 1977 [79]. It consists of the value n which is called modulus. The value e is called public exponent and the value d is called private exponent. The public key is defined as (n,e) and the private key is (n,d).

A standard way of generating the public key and the private key is by the following steps. First of all a pair of large prime numbers p and q are generated. The modulus number n is then calculated as the product of these two prime numbers, shown in Equation 2.

$$n = pg \tag{2}$$

The totient of n (let's call this *L(n)*) is then computed as Equation 3.

$$L(n) = (p - 1) * (q - 1) \tag{3}$$

Then the public exponent e is selected as a prime number between the value of 1 and *L(n)*. In other words, $1 < e < L(n)$. To find the private exponent d we use the modular inverse of e, shown in Equation 4. The modular inverse is solved by using the Extended Euclidean Algorithm.

$$d = e^{-1} mod\ L(n) \tag{4}$$

Once e and d have been found properly, the public key (n,e) and the private key (n,d) is ready to be used. The plaintext message m is encrypted with the public key as in Equation 5 and the ciphertext c is then decrypted with the private key as in Equation 6.

$$c = ENCRYPT(m) = m^e mod\ n \tag{5}$$

$$m = DECRYPT(c) = c^d mod\ n \tag{6}$$

RSA can also be used for signing messages to verify that the sender is legitimate. To do so, the sender (Alice) computes a hash function of the message and encrypts it with Equation 6 (c now represents the hashed message, the signature) using the private key. This is sent along with the original message as a signature. When the signed message arrives, the receiver (Bob) uses the same hash algorithm in conjunction with Alice's public key. Bob decrypts the signature (Equation 5) and compares the hash value with the message's actual hash value. This way Bob can verify that it is Alice (the legitimate author) who has sent the message. Another question is how Bob can trust that the public key has been generated from Alice's private key. This is done by key management, and the common key management scheme is Public Key Infrastructure (PKI) using X.509 certificate. In the X.509 system a Certificate Authority (CA) binds the public key to a specific name. By verifying that signature, the certificate can be trusted, and then also Alice's public key.

Another, faster way of signing messages is the Digital Signature Algorithm (DSA).

2.7.2 RSA - Security and key size

The inventors of RSA claim that a key of 1024-bits has been cracked by now. A key size of 2048-bits will be sufficient until 2030 according to current estimations. If high security level is needed beyond 2030, a 3078-bit key should be used. According to NSA even today, a 3078-bit key is recommended (for TOP-SECRET materials) and refers to [SP-800-56B Rev.1] [5]. A 768-bit key may still be used for lower valuable information, because it is still considered "hard-to-crack" according to RSA Laboratories [62].

2.7.3 Elliptic Curve Algorithms

The theory behind Elliptic Curve Cryptography (ECC) is that you calculate different points which give you the solution to a specific curve [76]. It works similar to the RSA algorithm (using modulus calculations) but contains more complex maths. In general, elliptic curve algorithms are fast and keep a high level of security with a smaller key size. The Diffie Hellman method is a well known Elliptic Curve Algorithm. The idea behind ECDH is that two starting points A and B are generated randomly (see Figure 31). Then a point (-C) is calculated with a line that goes through A and B. To get the third point, C, we simply take the reflection of -C. This method is called "addition" and means that C is the sum of A and B.

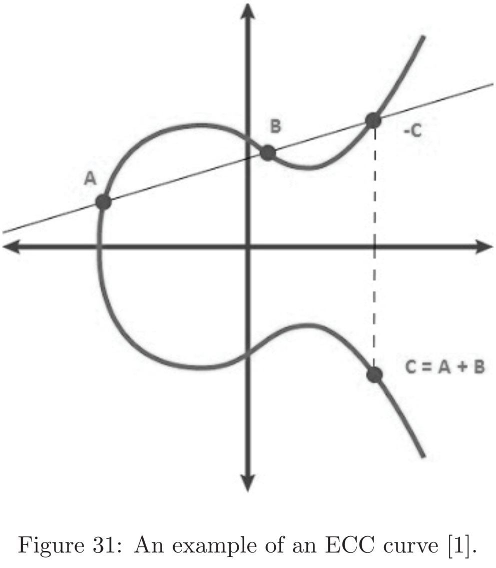


Figure 31: An example of an ECC curve [1].

Another method is "point doubling" and solves the issue of adding the same point twice (eg. A+A = 2A = C). It is very similar to the addition-method but instead of taking a line between A and B, the tangent of A is calculated and a point on the curve is given (-C). Reflect -C and the point C will be found. There is also a case when two points create a vertical line in order to find the third point, which will be infinite. That third point is called *point at infinity.*

Another group operation is the Scalar Multiplication, which means that adding the same point over and over again (point doubling many times) and corresponds to finding a integer to multiply the point with instead of using addition. An example of a ECC curve is the *Curve25519* which has the equation shown in Equation 7 (this is not the same curve as in figure 31).

$$y^2 = x^3 + 486662x^2 + x \tag{7}$$

This curve has been proved to be very fast and still remain a high level of security. Some other curves worth mentioning are the NIST P-224, P-256 and P-384. According to NIST [SP 800-78] these two curves are recommended for high level of security and NSA also seems to agree [5]. Users are skeptical to trust these two curves because they are not completely secure in the "Discrete Logarithm Problem" and due to other flaws (more on this in Chapter 2.7.4) [18].

Scalar multiplications on elliptic curves are basically one-way functions. A one-way function has the property of being relatively easy to compute an image of the elements in a domain but the image itself is difficult to reverse-engineer. Even if the starting point and the ending point on the curve are known, it is very hard to find how many calculations were made before reaching the final point. This is also referred to as a "Trapdoor function", which is considered as a great feature of the ECC algorithms. Some curves have easily been cracked though, and are thus now considered unsecure. A list of different curves (both secure and unsecure) can be found on SafeCurve's Website [18].

Elliptic curves can also be used for digital signatures, called Elliptic Curve Digital Signature Algorithm (ECDSA). It is simply a variant of DSA and uses elliptic curves. The advantage of ECDSA compared to DSA is that it is faster using smaller key size. According to Nick Sullivan at CloudFare (security for websites), a ECDSA certificate is up to 9.5 times faster than DSA when using only an eighth of the DSA key size [88].

2.7.4 ECC - Security and key size

The ECC requires less key-length compared to RSA with the same level of security. A 2048-bit RSA key would be comparable with a 224-bit ECC key, and a 3078-bit RSA key would be equal to a 256-bit ECC key. In other words an ECC curve would not need as much Central Processing Unit (CPU) power to complete its calculations and also require less Random Access Memory (RAM). This makes ECC algorithms very useful in mobile devices and smaller embedded system. In other words, the longer key size required the better ECC will be. According to Daniel Bernstein (a mathematician and cryptologist working at SafeCurves) and Tanja Lange (Professor at Technical University in Eindhoven) Curve25519 is the right choice for up to 192-bit security since it is faster compared to other curves and RSA [50]. For up to 256-bit security Curve41417 (also called Curve3617) is the right choice simply because it is even faster than Curve25519. For the highest security level, E-521 should be used.

According to Daniel Bernstein, many of the NIST curves have flaws [18]. An example is the NIST P-224 curve included in the [SP 186-4] which is known not to be twist secure. This NIST curve is slow, fragile, hard to test and hard to implement. This is where modern ECC curves come right in hand because they are faster and easier to implement. Examples of more modern curves that are considered secure are Curve25519 (from 2005) and Curve41417 (from 2013).

NSA have had influence in some of these matters with designing the NIST curves. This is based on Edward Snowden's articles about the NSA, more specifically how the NSA gather intelligence [82]. The New York Times publsihed an article in 2013 saying that some companies were coerced by the government to hand over their master encryption keys or building in back doors in their products [75]. Some of NSA's most intensive effort has also been focused on SSL, VPN and the protection used in the 4G smartphones. NSA efforts create a huge issue concerning the security in general. Bruce Schneier, a well known cryptographer, said that (based on some of the Snowden articles) NSA can decrypt most of the internet. They are doing it mainly by cheating, not by mathematics. As Bruce Schneier said: "Remember this: The math is good, but the math has no agency. Code has agency, and the code has been subverted.". NSA's work has created a backlash on the NIST curves, and there are other and better curves to use than the NIST curves.

2.8 Cryptographic libraries

Cryptographic libraries contain cryptographic schemes and functions for coding purposes. This chapter contains a description of three suitable libraries for the thesis.

2.8.1 mbdedTLS

Advanced RISC Machine (ARM) made the acquisition of the cryptographic "polarSSL" library in november 2014 [15]. After making the acquisition ARM decided to re-brand polarSSL as "mbdedTLS", with the focus of creating a cryptographic library specifically for an embedded system [16].

The cryptographic mbdedTLS library has a lack of documentation and a small user group compared to other more popular cryptographic libraries such as openSSL. Thus implementing and using the library functions is more time consuming than using for example the OpenSSL library.

2.8.2 Curve25519_donna

As mentioned earlier, ECC is good to use instead of RSA when fast computational time is desired. Dan J Bernstein (djb) developed the Curve25519 library in 2006 for fast key agreement [17]. The original implementation works only for 32-bit CPU's so a new one had to be developed for 64-bit CPU's. "Curve25519-donna-c64" was developed by Adam Langley (agl) to satisfy these needs. The usage is exactly the same one as djb's except for the function called "curve25519_donna".

In Table 6.2 of [SP 800-78] the key reference values are specified. The asymmetric cryptography schemes in this table are RSA (1024-bit or 2048-bit) and also the two curves P-256 or P-384. According to PIV requirements no less than 2048-bit keys are to be used with RSA. Even though curves like Curve25519 are faster, only these NIST curves are supported in the implementation of identifiers for supported cryptographic algorithms. However, agl's library is not very useful for the purpose of PIV cards.

2.8.3 OpenSSL

OpenSSL is the result of the ambition to create an open source library consisting of cryptographic tools. Currently the library is developed by a small team of volunteers in Europe and it is estimated that two-thirds of the Web-servers in the world use OpenSSL [46].

OpenSSL has a large user group, thus the documentation is rich and open-source code for projects that use OpenSSL is easy to find. Thus implementing and using the library functions is less time consuming than for example the mbdedTLS library.

2.9 Digital Certificates

This chapter will go through the purpose of Public Key Infrastructure, digital certificates, and why they are of interest for the thesis.

2.9.1 Public Key Infrastructure

A PKI is used to add public keys to entities, thus enabling verification of these entities in a system and providing key management in the system [60, p. 15]. The PKI architecture consists of the usage of certification authorities, registration authorities, public key cryptography and digital certificates.

Each certificate authority has two unique attributes, its public key and its name [60, p. 17]. The certificate authority creates and signs certificates and handles certificates they have issued, both those in use and those that have expired. When issuing a certificate the certificate authority signs the certificate with its private key and inserts its name. Thus a user can identify a certificate by its name and ensure authenticity by using the certificate authority's public key. The structure of a certificate issued by a certificate authority "A" is shown in Equation 8, where the private key is PR_{CA}, T is the time period the certificate is valid, PU_A is the public key and ID_A is A's name [57, p. 9].

$$C_A = E(PR_{CA}, [T, ID_A, PU_A]) \tag{8}$$

The registration authority is an entity that works as an intermediary between a certificate authority and the user that needs a certificate issued [57, p. 11].

The PKI architecture that is used by enterprises is generally either a mesh- or hierarchical structure [60, p. 19]. Both these structures are shown in Figure 32. The hierarchical structure consists of a certificate path that enables every certificate in the chain to be authenticated by simply following the certification path. Each entity in the hierarchical structure knows the public key of the root certificate authority. The mesh structure consists of independent certificate authorities that each issue a certificate to other certificates authorities in the mesh structure. Each entity in the mesh structure knows the public key of the entity that it is closest to.

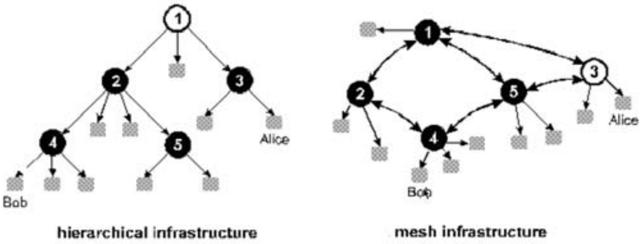
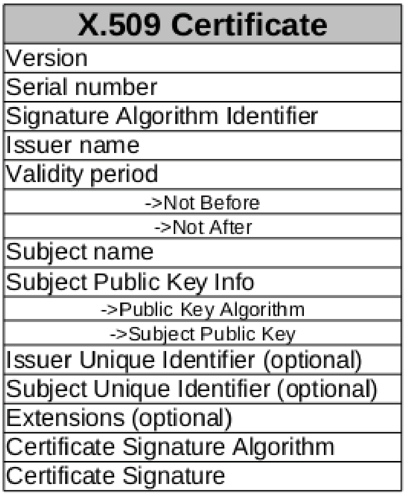


Figure 32: Two different PKI architectures [60, p. 19].

2.9.2 X.509 Certificate structure

The X.509 certificate was developed from the X.500 standard by International Telecommunication Union Telecommunication Standardization Sector (ITU-T) in 1988 [83]. The format of a X.509 certificate is shown in Figure 33. It contains a load of information such as the issuers name, validity period, public key information and much more. The mandatory fields are: the serial number, the signature algorithm identifier, the issuer name, the validity period, the subject name, and the public key information. The optional fields are the version number, the two issuer- and subject identifiers, and the extensions. These optional fields only appear in the certificate version 2 and 3. The "subjectPublicKeyInfo" field is of special interest for the verification process (mentioned in Chapter 4.1). This field contains information about cryptographic algorithm identifier, public key, and key size.



X.509 Certificate
Version
Serial number
Signature Algorithm Identifier
Issuer name
Validity period
->Not Before
->Not After
Subject name
Subject Public Key Info
->Public Key Algorithm
->Subject Public Key
Issuer Unique Identifier (optional)
Subject Unique Identifier (optional)
Extensions (optional)
Certificate Signature Algorithm
Certificate Signature

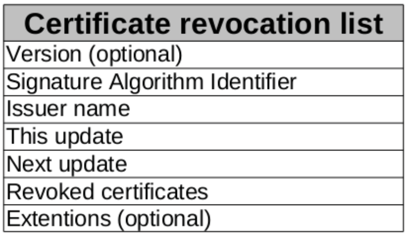
Figure 33: The X.509 (ver.3) Certificate structure.

Some information from the X.509 certificate is optional and the content in the mandatory fields may vary. It is important for the implementers to know what choices and consequences that a certain implementation gives, otherwise some choices may hinder interoperability.

The X.509 certificate is protected by a digital signature of the issuer. The Signature Algorithm ID field indicates which digital signature algorithm was used to protect the certificate. If the signature can by verified the certificate users know that its contents have not been tampered with and that the issuer actually is the trusted third party. Some certificates are more trustworthy than others based on what procedures used to issue them or what type of user cryptographic module that was used [60, p. 19].

2.9.3 Certificate revocation lists

In general certificates contain an expiry date but unfortunately certificates may become unreliable before the expiration date [60, p. 19]. One mechanism to update the status of the certificate is called *Certification Revocation List*. The Certification Revocation List CRL is protected by a digital signature of the CRL issuer. As for the X.509 certificate, if the CRL digital signature can be verified then its content can be trusted by its users.



Certificate revocation list
Version (optional)
Signature Algorithm Identifier
Issuer name
This update
Next update
Revoked certificates
Extentions (optional)

Figure 34: Certificate Revocation List format.

Many of the fields in the CRL have the same purpose as in a X.509 certificate. The main difference are the three fields *This update*, *Next update* and *Revoked certificates*. *This update* field indicates the issue date of the CRL. The *Next update* field indicates the date when the next CRL will be issued. The *Revoked Certificates* field lists the revoked certificates. The entry contains the certificate serial number, time of revocation and optional extensions. These entry extensions are used to provide additional information about the particular revoked certificate. Certificates may be revoked for a number of reasons. For example a specific cryptographic module may have been stolen or the module simply has been broken. The *reason code* extension describes why a specific certificate was revoked. A relying party may use this information to decide whether to accept or decline the previously generated signature.

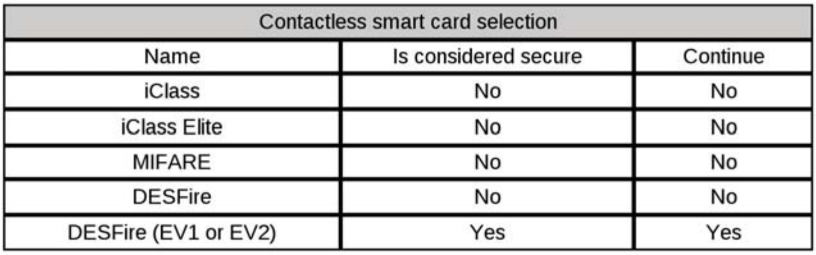
3 Evaluation

This chapter will go through the evaluation of this thesis work based on the theory and technical details in Chapter 2.

3.1 Evaluation of non FIPS-201 Implementations

Using the data obtained from investigating different implementation alternatives that are not dependent on for example CIV or PIV compatibility (Chapter 2.3 and 2.4) a conclusion can be drawn. As mentioned in Chapter 2.4, iClass, iClass Elite, MIFARE and DESFire are not considered secure today, but DESFire EV1 and EV2 are.

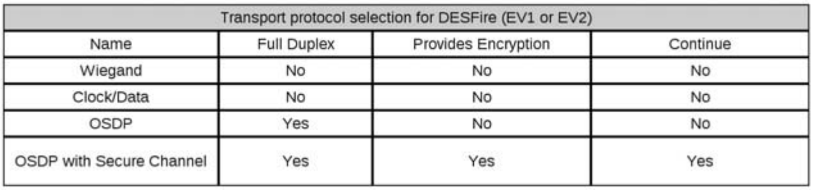
The common divider among the transport protocols are that they need support for half duplex mode, which is not available for Wiegand and Clock/Data as they use simplex. TCP/IP could be of interest but due to lack of documentation when implementing the cryptographic schemes, OSDP (with Secure Channel) seems more appealing. Also, the Axis A1001 Network Door Controller already has full support for OSDP over RS485. Another disadvantage with TCP/IP is that the card readers need to be replaced in the current solution (see Chapter 3.5). As shown in Figure 35 and Figure 36 this implementation is a solution combining DESFire (EV1 or EV2) and OSDP (with Secure Channel).



Contactless smart card selection

Name	Is considered secure	Continue
iClass	No	No
iClass Elite	No	No
MIFARE	No	No
DESFire	No	No
DESFire (EV1 or EV2)	Yes	Yes

Figure 35: Contactless Smart Card selection.



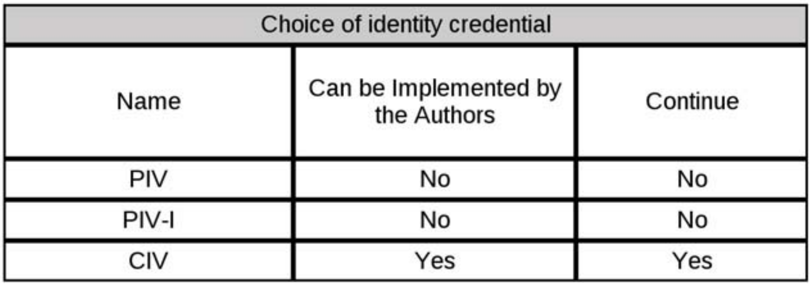
Transport protocol selection for DESFire (EV1 or EV2)

Name	Full Duplex	Provides Encryption	Continue
Wiegand	No	No	No
Clock/Data	No	No	No
OSDP	Yes	No	No
OSDP with Secure Channel	Yes	Yes	Yes

Figure 36: Transport protocol selection for DESFire (EV1 or EV2).

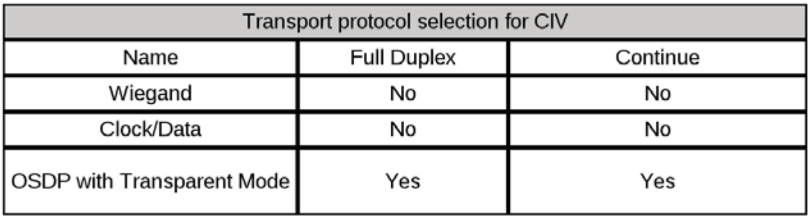
3.2 Evaluation of FIPS-201 Implementations

Using the data obtained from investigating different implementation alternatives (Chapter 2.3 and 2.5) a conclusion can be drawn. Both PIV and PIV-I require government cooperation and background checks which complicates product development in the short term. A CIV implementation is of more interest since Axis Communications can implement CIV solutions themselves without government influence and keep a high level of security for their credentials. As shown in Figure 37 and Figure 38 this implementation is a solution combining CIV and OSDP (with Transparent Mode).



Choice of identity credential		
Name	Can be Implemented by the Authors	Continue
PIV	No	No
PIV-I	No	No
CIV	Yes	Yes

Figure 37: Choice of identity credential.



Transport protocol selection for CIV		
Name	Full Duplex	Continue
Wiegand	No	No
Clock/Data	No	No
OSDP with Transparent Mode	Yes	Yes

Figure 38: Transport protocol selection for CIV.

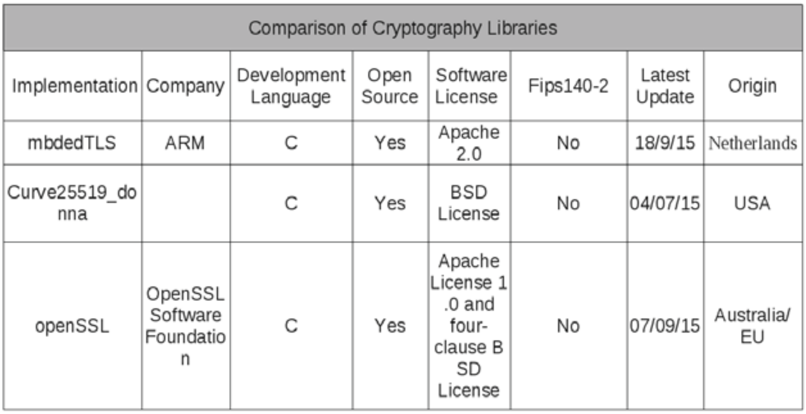
3.3 Evaluation of CIV and Derived CIV

The CIV implementation currently makes Axis dependent on smart card issuers while the Derived CIV implementation makes Axis non-dependent on smart card issuers as the Derived CIV implementation can be implemented using a smartphone on the credential side instead of a smart card. Therefore Derived CIV is to be implemented in the solution for this project to explore all technical options. The physical smart card solution will be left out for future work.

3.4 Evaluation of Cryptographic libraries

The following evaluation is based on the contents of Chapter 2.8. Based upon which Algorithms and Key Sizes that are required for PIV key types and the functionality and ease of use of the Cryptographic library, OpenSSL was chosen.

More precisely the user group of mbdedTLS is too small while Curve25519_donna did not support any of the required PIV algorithms.

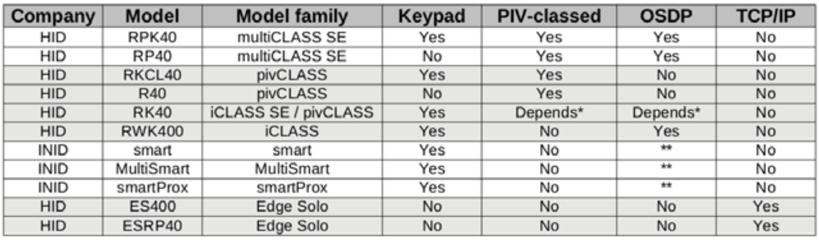


Comparison of Cryptography Libraries							
Implementation	Company	Development Language	Open Source	Software License	Fips140-2	Latest Update	Origin
mbdedTLS	ARM	C	Yes	Apache 2.0	No	18/9/15	Netherlands
Curve25519_donna		C	Yes	BSD License	No	04/07/15	USA
openSSL	OpenSSL Software Foundation	C	Yes	Apache License 1.0 and four-clause BSD License	No	07/09/15	Australia/ EU

Figure 39: Comparison of the Cryptography Libraries.

3.5 Card reader comparison

There are plenty of card readers on the market to choose from, but the common denominator is that the card readers need support for OSDP (ver.2). OSDP offers Secure Channel Mode and Transparent Mode which is of interest to Axis Communications when implementing higher security level on the Axis Communications A1001 Network Door Controller. For high security credentials they must also support PIV card standards.



Company	Model	Model family	Keypad	PIV-classed	OSDP	TCP/IP
HID	RPK40	multiCLASS SE	Yes	Yes	Yes	No
HID	RP40	multiCLASS SE	No	Yes	Yes	No
HID	RKCL40	pivCLASS	Yes	Yes	No	No
HID	R40	pivCLASS	No	Yes	No	No
HID	RK40	iCLASS SE / pivCLASS	Yes	Depends*	Depends*	No
HID	RWK400	iCLASS	Yes	No	Yes	No
INID	smart	smart	Yes	No	**	No
INID	MultiSmart	MultiSmart	Yes	No	**	No
INID	smartProx	smartProx	Yes	No	**	No
HID	ES400	Edge Solo	No	No	No	Yes
HID	ESRP40	Edge Solo	No	No	No	Yes

Figure 40: A comparison between different Card Readers. ∗ The RK40 reader comes in two classifications, iCLASS SE and pivCLASS [45]. This means that the RK40 can be compatible with PIV cards if it is pivCLASS but do not support OSDP with the same classification (OSDP needs iCLASS SE classification). ∗∗ INID has implemented OSDP support in some of their readers but it is not considered bug-free [13].

There are a few interesting card readers in Figure 40. The need of both OSDP and PIV compatibility reduces the potential candidates to only two, RP40 and

RPK40. When comparing these two it all comes down to whether keypad is needed or not. Both these readers are compatible with PIV, PIV-I and CIV cards which is good for alternative solutions for higher credential security.

3.6 Summary of the evaluations

The following figures shows the evaluation of techniques and solutions described in Chapter 3.1 and 3.2. The final conclusion of these evaluations is that the Derived CIV solution combined with OSDP Transparent Mode will be used for implementation in this project (see Figure 46). The solution combining DESFire EV1 and OSDP Secure Channel (see Figure 43) will be left out for future work. Even the solution with a physical CIV classed smart card is left out for future work. HID's RPK40 multiClass SE was chosen for the solution as it supports OSDP (ver.2) and also includes a keypad which is mandatory for pin code usage.

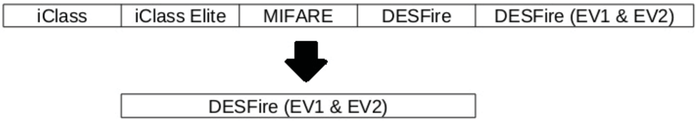


Figure 41: Evaluation of the contactless smart card technologies.

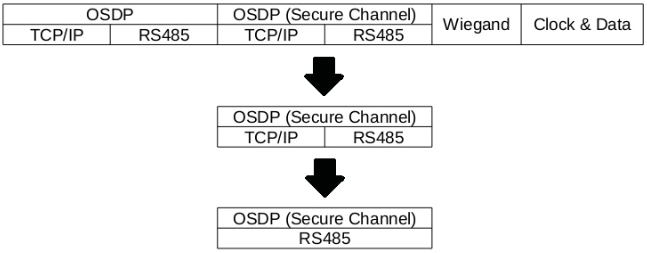


Figure 42: Evaluation of the transport protocols between the card reader and the Axis A1001 Network Door Controller.

.

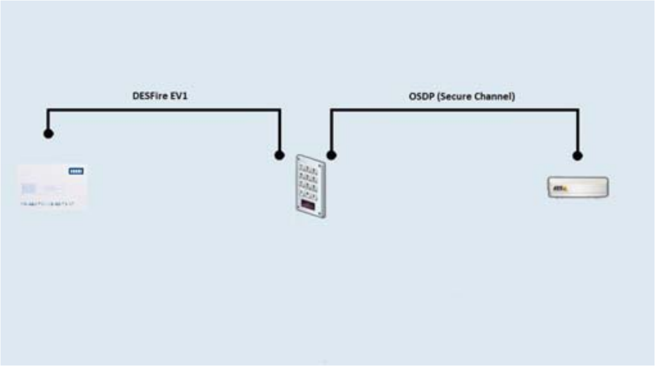


Figure 43: Structure of solution using DESFire (EV1 or EV2) and OSDP Secure Channel.

.

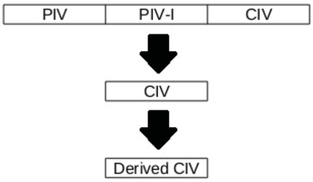


Figure 44: Evaluation of the FIPS-201 identity credentials.

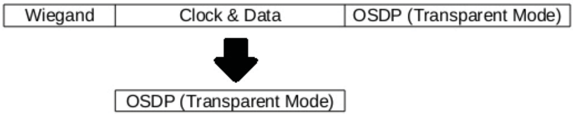


Figure 45: Evaluation of the transport protocols between the smartphone, card reader and the Axis A1001 Network Door Controller.

.

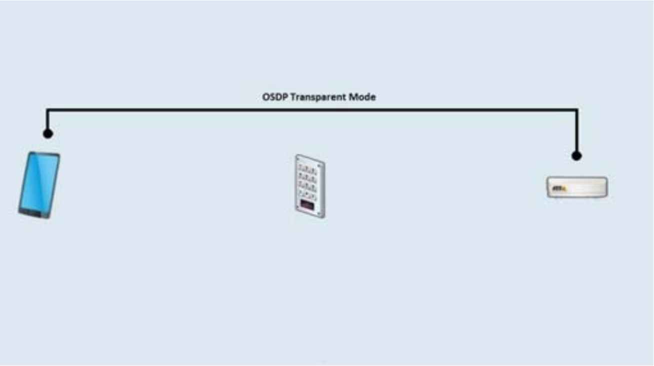


Figure 46: Structure of solution using derived CIV and OSDP Transparent Mode.

.

4 Implementation

This chapter will go through how the implementation has been done based on the evaluation in Chapter 3. Some technical details have been left out due to confidentiality.

4.1 PIV Authentication

Authentication mechanisms using the asymmetric Card Authentication Key are shown in Figure 47. The ”PIV Application on Local System” corresponds to the Axis Communications A1001 Network Door Controller. The communication between the A1001 and the PIV credential holder is implemented with the OSDP functionality known as ”Transparent Mode”.

After setting up the communication between the A1001 and the PIV credential holder, the trusted party (A1001) will request a certificate from the card to be able to verify the trustworthiness. This certificate is in X.509 format and contains a subjectPublicKeyInfo field where the public key, information about key size and the PIV algorithm identifier is stored. This information will be used by the A1001 and in the next step it will request a card signature by creating a nonce and send it to the PIV credential holder. The cardholder then signs the nonce and sends it back to the A1001. To verify this signature the A1001 uses the public key from the received X.509 certificate, thus protecting itself from a replay attack as the nonce will only be sent to the controller once. After this process the cardholder can be granted access to the requested system.

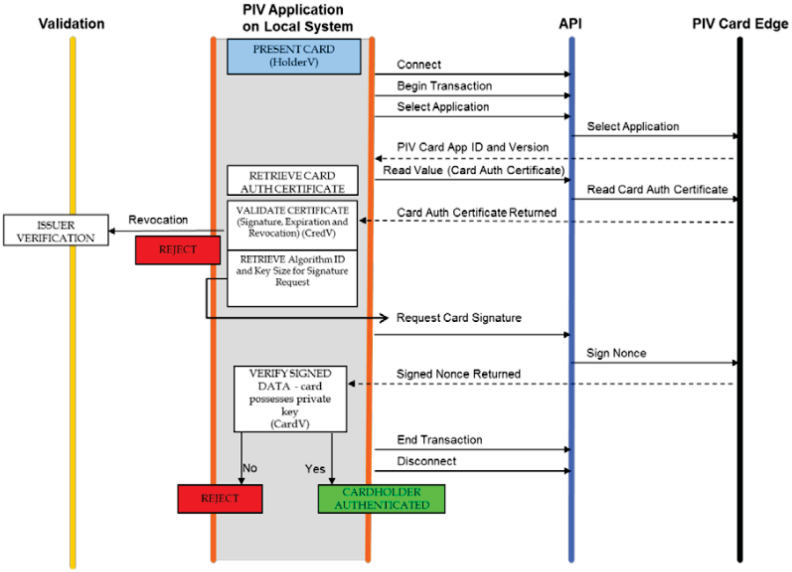

Validation
PIV Application on Local System
API
PIV Card Edge
PRESENT CARD (HolderV)
Connect
Begin Transaction
Select Application
Select Application
PIV Card App ID and Version
RETRIEVE CARD AUTH CERTIFICATE
Read Value (Card Auth Certificate)
Read Card Auth Certificate
VALIDATE CERTIFICATE (Signature, Expiration and Revocation) (CredV)
Card Auth Certificate Returned
ISSUER VERIFICATION
Revocation
REJECT
RETRIEVE Algorithm ID and Key Size for Signature Request
Request Card Signature
Sign Nonce
VERIFY SIGNED DATA - card possesses private key (CardV)
Signed Nonce Returned
End Transaction
Disconnect
No
Yes
REJECT
CARDHOLDER AUTHENTICATED

Figure 47: Authentication using an asymmetric Card Authentication Key [71]. PIV Application on Local System corresponds to the Axis A1001 Network Door Controller, API corresponds to the HID card reader, and the PIV Card Edge corresponds to the Android smartphone.

4.2 APDU structure

In this project Android was chosen as the platform to work on. Android has built-in support for handling Application Protocol Data Unit (APDU) data packages, a class called *HostApduService* [30]. *HostApduService* is an abstract public class in Java which basically emulates a card with a NFC implementation. When a remote NFC device wants to talk to a specific device that uses the *HostApduService* class, it needs to be paired with a specific Application Identifier (AID). This is done with a so-called ”SELECT FILE” APDU command (read more in Chapter 4.2.5). The ”SELECT FILE” is based on the ISO 7816-4 standard [22]. When implementing the *HostApduService* class the minimum requirements will be set to the API level 19 (Android 4.4 Kitkat) or newer in order to use the application.

4.2.1 APDU package specifications

The APDU command package contains a header and a body (see Figure 48). The command header includes flags and identifiers which indicate what commands are sent and what information the data field contains. The command body contains information about the outgoing data field and expected data length in the response. According to ISO/IEC 7816 the parameters **P1** and **P2** are instruction parameters. In [SP 800-73-4] these two parameters are the Algorithm reference respectively the Key reference when using the general authenticate card command. Below is an overview and explanation of the different parts of the command package [21].

CLA - Class instruction byte.
INS - Instruction byte. Defines what instruction is sent.
P1 - Instruction parameter 1
P2 - Instruction parameter 2
L$_c$ - Number of bytes in the data field of the command.
Data field - Data field bytes.
L$_e$ - Maximum number of bytes expected in the data field of the response to the command.

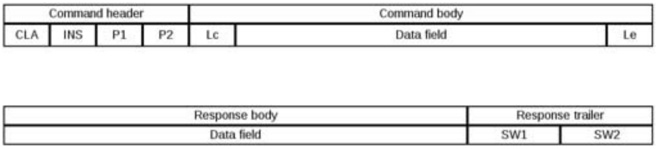


Figure 48: Command and response structure for APDU.

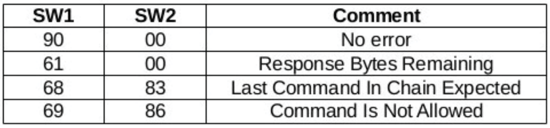
.

The APDU response package contains a body and a trailer (see Figure 48). The response body contains the data field in the response. The response trailer contains status information. For example when a successful execution is done the **SW1 SW2** will return '90 00'. Below is an overview and explanation of the different parts of the response package.

Data field - Data field bytes. For TLV-coded data fields, SIMPLE-TLV and BER-TLV coded data objects are supported by ISO/IEC 7816.
SW1 - Status byte 1
SW2 - Status byte 2

4.2.2 Status response

As mentioned in Chapter 4.2.1 the fields **SW1** and **SW2** will handle status messages. These responses are sent along with every APDU response. The existing command-response cases does not affect the status response. In other words, the response always contains a minimum of 2 bytes (the response trailer in Figure 48) whatever the response case might be. There are a lot of predefined status messages in ISO 7816-4. The status messages used in this project can be seen in Figure 49.



SW1	SW2	Comment
90	00	No error
61	00	Response Bytes Remaining
68	83	Last Command In Chain Expected
69	86	Command Is Not Allowed

Figure 49: Status messages used in the implementation

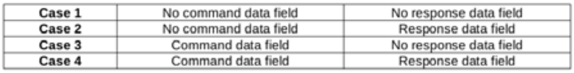
.

4.2.3 Command chaining

Command chaining is used when the length of the message is longer than the length of the data field [71, p. 17]. Command chaining divides the message into smaller parts thus enabling it to be sent in multiple parts and its parts assembled correctly as it is received. When command chaining is used **INS** is set to '87' and **CLA** is set to '00' or '10'.

4.2.4 Command/Response cases

The command-response can have four different cases as shown in Figure 50, and more sub-cases for that matter [20]. What case to use depends on what command is sent and this is handled by the APDU. For example when sending a chained message (mentioned in Chapter 4.2.3) the first chain will be case 1 and the last chain will use case 4, simply because there is no requested data in the first response.



Case 1	No command data field	No response data field
Case 2	No command data field	Response data field
Case 3	Command data field	No response data field
Case 4	Command data field	Response data field

Figure 50: Four different command cases that are used by the APDU.

.

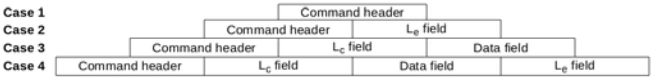


Figure 51: Detailed view of the four different command cases that are used by the APDU.

.

4.2.5 PIV Card AID

For a PIV card implementation the AID needs to be specified in the "SELECT FILE" command that *hostApduService* class is using based on the ISO 7816-4 standard. The Application IDentifier (AID) of a PIV card is specified in [SP-800-73]. The following AID required for PIV application is:

'A0 00 00 03 08 00 00 10 00 10 00'

The first part of the AID ('A0 00 00 03 08') is the NIST RID, followed by NIST PIX PIV Card Application ('00 00 10 00') and NIST PIX version portion ('10 00') indicating the first PIV Card Application. The selection of the PIV Card Application can either be with the full sequence above or without the version portion, as follows:

'A0 00 00 03 08 00 00 10 00'

4.3 Cryptography using the Android API

When implementing the CIV credentials in a mobile application, the cryptographic requirements (see Figure 19) are already implemented in the Android SDK from API level 1. Over the past years Android has developed a lot in these matters and offers more features and new classes in the cryptographic areas such as new cryptographic schemes, new hash functions, setting padding option for encryption, and much more. Some of these new features require API level 21 (Android 5.0 Lollipop) and some require API level 23 (Android 6.0 Marshmallow). An example of this is the class *KeyProperties*, which is a class where you can modify a lot of properties for the keys stored in Android *KeyStore*. Another example is the *KeyGenParameterSpec* which lets you change the settings (for example a padding scheme) for key generation, which was added in API level 23.

5 Results

This chapter will go through the results of the thesis work based on the evaluation in Chapter 3 and implementation in Chapter 4.

5.1 Implementation of a Derived CIV authentication protocol

The basis of the PIV authentication protocol (shown in Figure 47) and the Derived CIV solution combined with OSDP Transparent Mode (shown in Figure 46) was used to structure an authentication protocol as shown in Figure 52.

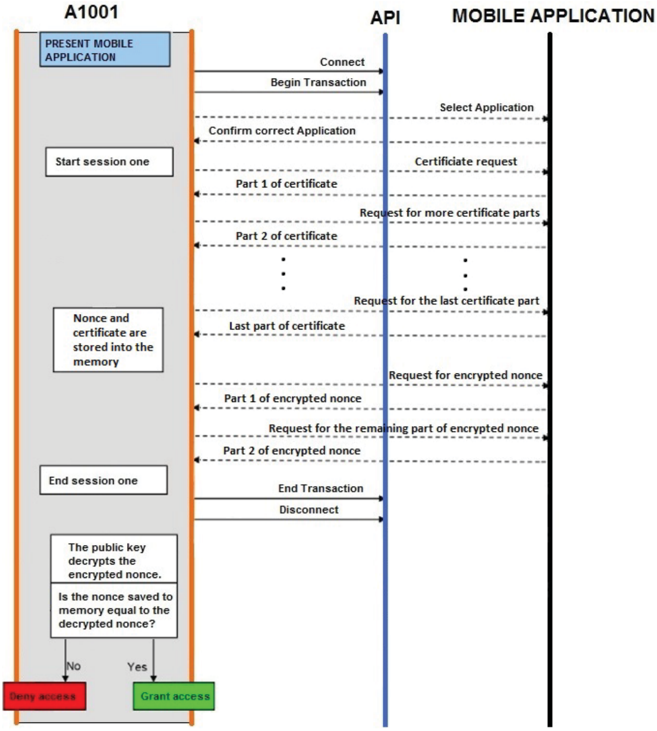

A1001
API
MOBILE APPLICATION
PRESENT MOBILE APPLICATION
Connect
Begin Transaction
Select Application
Confirm correct Application
Start session one
Certificiate request
Part 1 of certificate
Request for more certificate parts
Part 2 of certificate
Request for the last certificate part
Nonce and certificate are stored into the memory
Last part of certificate
Request for encrypted nonce
Part 1 of encrypted nonce
Request for the remaining part of encrypted nonce
Part 2 of encrypted nonce
End session one
End Transaction
Disconnect
The public key decrypts the encrypted nonce.
Is the nonce saved to memory equal to the decrypted nonce?
No
Yes
Deny access
Grant access

Figure 52: Authentication Protocol with certificates based on the CIV authentication protocol

.

An issue arose during the process of implementing the command chaining of the certificate. After a small amount of chains the HID RK40 reader stopped the command chaining process by not sending any more APDU command packages to the smartphone device, the HID RK40 instead replied with an error message. After attempting to debug the code both on the A1001 and smartphone it was decided due to time constraints for the Master Thesis that a different structure was to be implemented, enabling a smaller manageable data quantity for the command chaining process. The aforementioned structure is described in Chapter 5.2.

5.2 Implementation based on the Derived CIV authentication protocol

The issue described in Chapter 5.1 generated an alternative solution based upon the CIV authentication protocol shown in Figure 52. The command chaining of the X.509 certificate was circumvented by instead command chaining a public key generated and sent by the smartphone. Thus the new solution shown in Figure 53 enabled a smaller manageable data quantity for the command chaining process, and was also able to be implemented in code.

5.2.1 Execution times

When using the application, the execution time for the first session was around 400 ms and for the second session it was around 450-500ms.

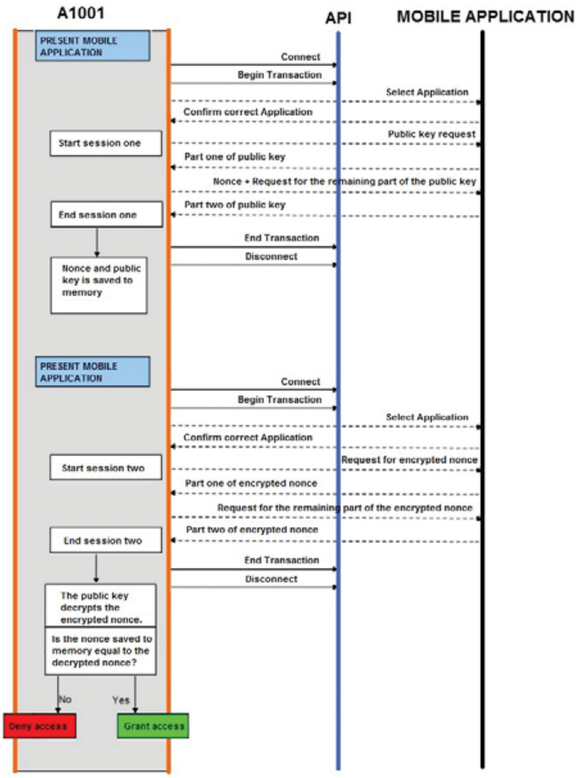

A1001
API
MOBILE APPLICATION
PRESENT MOBILE APPLICATION
Connect
Begin Transaction
Select Application
Confirm correct Application
Public key request
Start session one
Part one of public key
Nonce + Request for the remaining part of the public key
Part two of public key
End session one
End Transaction
Disconnect
Nonce and public key is saved to memory
PRESENT MOBILE APPLICATION
Connect
Begin Transaction
Select Application
Confirm correct Application
Request for encrypted nonce
Start session two
Part one of encrypted nonce
Request for the remaining part of the encrypted nonce
Part two of encrypted nonce
End session two
End Transaction
Disconnect
The public key decrypts the encrypted nonce.
Is the nonce saved to memory equal to the decrypted nonce?
No
Yes
Deny access
Grant access

Figure 53: Authentication Protocol with public key based on the CIV authentication protocol

.

6 Conclusion

The transport protocols that were of interest for the supervisors from Axis Communications are described in Chapter 2.3 and Chapter 2.4. Those transport protocols are the ones currently of interest for the Physical Access Control System (PACS) industry, a large number of them are thus supported by Axis Communications PACS products. The information that laid the foundation for writing about the different transport protocols were either from scientific papers, the industry or people from within the industry.

A large amount of the Transport protocols on the market are not considered secure, something that is also true for a large amount of the different protocols in Chapter 2.3 and Chapter 2.4. An important conclusion was drawn from Chapter 2.3: the importance of half duplex communication to establish a secure tamper-resistant connection between a card reader and a PACS system, a conclusion that had a major impact on the evaluation process in Chapter 3. This thesis was in cooperation with Axis Communications which had an effect on the evaluation process in Chapter 3, since the chosen transport protocol needed to be supported by Axis Communications PACS products.

One of the goals with this thesis was the analysis and evaluation of various authentication and authorization techniques with a high level of safety for smart cards and other mobile devices. The two candidates that were suggested for investigation by the supervisors at Axis Communications were PLAID and PIV. As described in Chapter 2.5.7, PLAID is considered unsafe and was thus not of interest to be implemented in this thesis.

The PIV solutions used in the U.S. have been very successful. Companies in the private sector have been looking at these solutions in order to offer more secure products to their costumers. The problem with the PIV solution in the private sector is that it requires government influence and backgrounds checks. This is where CIV comes in. As described in Chapter 2.5.4 CIV does not require any background checks, no influence from the government but is recommended to follow [SP 800-73] in its implementation. This is what is more interesting for companies in the private sector. The CIV still keeps a very high security level for its smart card implementation. By implementing CIV, a high security end-to-end solution is obtained.

Gaining an understanding for different cryptographic schemes and methods was of importance since implementing a secure end-to-end solution was a major part of this thesis. The information about different cryptographic schemes described in Chapters 2.6 and 2.7 gives an understating of which schemes fit the implementation phase and which do not. The conclusion drawn from Chapter 2.5.1 is that either RSA or ECC are to be used in the implementation. When implementing cryptography one should strive to use open-source solutions that have been tested by a multitude of different security specialists, Chapter 2.8 lays the foundation for the evaluation of different Cryptographic libraries done in Chapter 3.4. As Chapter 3.4 states, the conclusion is that the OpenSSL Cryptographic library is to be used in the implementation, as this has the largest user group and contains the PIV required algorithms.

A major goal with this thesis was the prototyping of access control with high security credentials on an embedded system in combination with a smartphone. The evaluation described in Chapter 3 laid the foundation for the basis of the implementation described in Chapter 4 and the result of the implementation in Chapter 5. As Chapter 5 describes, the coding process was more of an agile process than a waterfall based one. This allowed for major changes during the process of the prototyping. One major change was to the structure of the solution as described in Chapter 5.1 and 5.2, the result of an error also described in those chapters.

The implemented solution as described in Chapter 5.2 chains the data twice thus not enabling it to send a X509 certificate, as the certificate would need to be chained multiple times. Since an X509 certificate is not utilized the implemented solution does not fulfill the qualification to make it a Derived CIV authentication protocol. The implemented solution does, however, lay a foundation for a functional Derived CIV authentication protocol (described in Chapter 5.1), this is more thoroughly discussed in Chapter 6.1. The current implementation only supports RSA encryption. Implementing ECC encryption also remains a possibility. This is more thoroughly discussed in Chapter 6.1.

6.1 Future work

As explained in Chapter 2.5.5 storing credentials (in this thesis case a private key) in a software-based HSM is not recommended. Thus a future solution to develop is how to store the private key in a safe environment on the users smartphone, preferably as stated in Chapter 2.5.5 on a Trusted Platform Module, a Trusted Execution Environment or a Secure Element as an embedded hardware-based HSM. The private keys could also be stored in a removable HSM but as also stated in Chapter 2.5.5 the market has a lack of interest in that solution.

In the implemented solution the specific PIV AID were not set properly, thus not making it recognizable as a PIV card. Therefore this task is put for future work in the implementation.

Another idea proposed during the thesis was the combination of DESFire EV1 and OSDP Secure Channel (read the proposed solution in Chapter 3.6 and see Figure 43). This may also be of interest in the future and was listed as a future work based on the gathered information in this thesis. Also, there is the possibility for companies to issue their own physical CIV card. With reference to this thesis groundwork (as a derived solution was chosen) this was left out for future work as well.

The most important part is the "failed" command chaining due to hardware issues. As the most important future work, this must be solved in order to be able to implement a real CIV card in the smartphone. As the solution currently implemented lays a foundation for a functional Derived CIV authentication protocol (described in Chapter 5.1), implementing the Derived CIV authentication protocol is mainly a matter of debugging the hardware issues. Debugging the hardware issues was something the authors of this thesis (due to time constraints for the thesis) were not able to do.

The last detail in the future work section is the support for ECC encryption (NIST P-256 and NIST P-384). This has not been implemented yet. The reason for implementing this is not only to support both encryption methods, but also for testing the difference in execution time between the cryptographic schemes. This would have made a very interesting aspect of the thesis since one of the questions were about work flows, and also since ECC is known for being faster than RSA when it comes to encryption. For the current solution with RSA encryption, the execution time in the smartphone applitcation was around 850-900 ms in total (read more in Chapter 5.2.1). Something that could have had an effect on user management, as a time reduction might have been attained.

References

[1] curveplot.gif. `http://www.hpl.hp.com/research/info\_theory/images/curveplot.gif`, 2015. Online: Accessed 30 Oct 2015.

[2] Ecb_encryption.svg, ecb_decryption.svg. `https://en.wikipedia.org/wiki/Block\_cipher\_mode\_of\_operation\#Common\_modes`, 2015. Online: Accessed 30 Oct 2015.

[3] ASSA ABLOY. Odsp compliant r15, r30, r40, rk40, rkl55 iclass readers. *HID Corporation*, 2015. Accessed 30 Nov 2015.

[4] Advantech. Basics of the rs-485 standard. `http://www.bb-elec.com/Learning-Center/All-White-Papers/Serial/Basics-of-the-RS-485-Standard.aspx`. Online: Accessed 30 Nov 2015.

[5] National Security Agency. Cryptography today. `https://www.nsa.gov/ia/programs/suiteb\_cryptography/index.shtml`, 2015. Online: Accessed 25 Oct 2015.

[6] Smart Card Alliance. Ficam in brief: A smart card alliance summary of the federal identity, credential, and access management (ficam) roadmap and implementation guidance. 2010. Accessed 15 Oct 2015.

[7] Smart Card Alliance. Personal identity verification interoperability (piv-i) for non-federal issuers: Trusted identities for citizens across states, counties, cities and businesses. *Smart Card Alliance*, 2011. Accessed 17 Oct 2015.

[8] Smart Card Alliance. A”comparison”of”piv,”piv1i”and”civ”credentials. *smart Card Alliance*, 2012. Accessed 15 Oct 2015.

[9] Gabriella Arcese, Giuseppe Campagna, Serena Flammini, and Olimpia Martucci. Near field communication: Technology and market trends. *Department of Business Studies, Roma Tre University, Rome, Italy*, 2014. Accessed 6 Nov 2015.

[10] IEEE Standards Association. Ieee standard 802.3at-2009. 2009. Accessed 1 Dec 2015.

[11] Security Industry Association. Open supervised device protocol (osdp) version 2.1.5. 2012. Accessed 2 Dec 2015.

[12] Security Industry Association. Sia osdp frequently asked questions. 2015. Accessed 30 Nov 2015.

[13] Björn Hall at Axis Communication, 2015. 8 Nov 2015.

[14] Michael Backes. Lecture notes for cs-578 cryptography (ss2007). *Saarland University*, 2007. Accessed 3 Nov 2015.

[15] Paul Bakker. Polarssl is now a part of arm. `https://tls.mbed.org/tech-updates/blog/polarssl-part-of-arm`, 2014. Online: Accessed 12 Nov 2015.

[16] Paul Bakker. mbed tls 1.3.10 released. `https://tls.mbed.org/tech-updates/releases/mbedtls-1.3.10-released`, 2015. Online: Accessed 12 Nov 2015.

[17] Daniel J. Bernstein. curve25519-donna. `https://code.google.com/p/curve25519-donna/`. Online: Accessed 12 Nov 2015.

[18] Daniel J. Bernstein and Tanja Lange. Safecurves: choosing safe curves for elliptic-curve cryptography. `http://safecurves.cr.yp.to`, 2014. Online: Accessed 25 Oct 2015.

[19] Lammert Bies. Rs485 serial information. `http://www.lammertbies.nl/comm/info/RS-485.html\#intr`, 2015. Online: Accessed 29 Nov 2015.

[20] CardWerk. Iso 7816-4: Annex a: Transportation of apdu messages by t=0. 2015. Accessed 28 Nov 2015.

[21] CardWerk. Iso 7816-4: Interindustry commands for interchange, 5. basic organizations. 2015. Accessed 28 Nov 2015.

[22] Cardwerk. Iso 7816-4: Interindustry commands for interchange, 6. basic interindustry commands. 2015. Accessed 28 Nov 2015.

[23] U.S. Department Of Commerce. Fips pub 201-2 personal identity verification (piv) of federal employees and contractors. *Computer Security Division Information Technology Laboratory*, 2013. Accessed 4 Dec 2015.

[24] Axis Communication. Axis q6032-e ptz dome network camera datasheet. Technical report. Accessed 1 Dec 2015.

[25] Axis Communication. Axis a1001 network door controller datasheet. Technical report, 2014. Accessed 2 Dec 2015.

[26] Axis Communication. Om axis communication. `http://www.axis.com/se/sv/about-axis`, 2015. Online: Accessed 27 Nov 2015.

[27] David Cooper, Hidlegard Ferraiolo, Ketan Mehta, Salvatore Francomacaro, Ramaswamy Chandramouli, and Jason Mohler. Nist special publication 800-78-4. *National Institute of Standards and Technology*, 2015. Accessed 3 Dec 2015.

[28] Francisco Corella. Nist fails to address concerns on derived credentials. http://pomcor.com/2015/01/30/nist-fails-to-address-concerns-on-derived-credentials/, 2015. Accessed 21 Nov 2015.

[29] Jean Paul Degabriele, Victoria Fehr, Marc Fischlin, Tommaso Gagliardoni, Felix Günther, Giorgia Azzurra Marson, Arno Mittelbach, and Kenneth G. Paterson. Unpicking plaid a cryptographic analysis of an iso-standards-track authentication protocol. *Information Security Group, Royal Holloway, University of London, U.K. and Cryptoplexity, Technische Universität Darmstadt, Germany*, 2015. Accessed 18 Oct 2015.

[30] Android Developer. Hostapduservice. `http://developer.android.com/reference/android/nfc/cardemulation/HostApduService.html`, 2015. Online: Accessed 28 Nov 2015.

[31] Tony Diodato and Joe Gittens. Youtube: Webinar: How does the osdp standard affect my company? `https://www.youtube.com/watch?v=upUUssZbvoE`. Online: Accessed 15 Sept 2015.

[32] Inc DSX Access Systems. What is clock and data. `http://www.dsxinc.com/designguide2/docs2/whatisclockanddata.pdf`, 2011. Accessed 29 Nov 2015.

[33] Morris Dworkin. Nist special publication 800-38a recommendation for block cipher modes of operation. *National Institute of Standards and Technology*, 2001. Accessed 30 Oct 2015.

[34] Information Sharing Environment. Federal identity credential and access management (ficam) maturity model. `https://www.ise.gov/federal-identity-credential-and-access-management-ficam-maturity-model`. Online: Accessed 15 Oct 2015.

[35] Hidlegard Ferraiolo, David Cooper, Salvatore Francomacaro, Andrew Regenscheid, Jason Mohler, Sarbari Gupta, and William Burr. Guidelines for derived personal identity verification (piv) credentials. *National Institute of Standards and Technology*, 2014. Accessed 16 Nov 2015.

[36] FingerTec. Tcp/ip network as main communication in eletronic access control systems. 2013. Accessed 30 Nov 2015.

[37] Flavio D. Garcia, Gerhard de Koning Gans, Roel Verdult, and Milosch Meriac. Dismantling iclass and iclass elite. *Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands and Bitmanufaktur GmbH, Germany.* Accessed 23 Nov 2015.

[38] Gartner. Gartner hype cycle. `http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp`, 2015. Online: Accessed 8 Nov 2015.

[39] gemalto. Idprime certificate-based smart cards and secure devices. 2015. Accessed 1 Nov 2015.

[40] Kevin Gingerich. Rs-485 unit load and maximum number of bus connections. *Texas Instruments*, 2014. Accessed 29 Nov 2015.

[41] HID Global. Iso/iec 7498-1 information technology - open systems interconnection - basic reference model: The basic model. *International Standard*, 1996. Accessed 28 Nov 2015.

[42] HID Global. Hid clock-and-data reader output. 1998. Accessed 29 Nov 2015.

[43] HID Global. Understanding card data formats. 2006. Accessed 28 Nov 2015.

[44] HID Global. Actividentity's cms appliance adds smart card alliance's new civ credential to long list of capabilities. `http://www.hidglobal.com/press-releases/actividentitys-cms-appliance-adds-smart-card-alliances-new-civ-credential-long-list`, 2012. Online: Accessed 17 Oct 2015.

[45] HID Global. Readers. `http://www.hidglobal.com/products/readers`, 2015. Online: Accessed 8 Nov 2015.

[46] Dan Goodin. Critical crypto bug in openssl opens two-thirds of the web to eavesdropping. 2014. Accessed 14 Okt 2015.

[47] Matthew Green. Attack of the week: Unpicking plaid. `http://blog.cryptographyengineering.com/2014/10/attack-of-week-unpicking-plaid.html`, 2014. Online: Accessed 19 Oct 2015.

[48] Security Equipment Integration Working Group. Review of the open supervised device protocol (osdp) for dod applicability. 2014. Accessed 30 Nov 2015.

[49] The TCP/IP Guide. Simplex, full-duplex and half-duplex operation. `http://www.tcpipguide.com/free/t\_SimplexFullDuplexandHalfDuplexOperation.htm`, 2005. Online: Accessed 30 Nov 2015.

[50] HackersOnBoard. Youtube: Shmoocon 2014: Safecurves: Choosing safe curves for elliptic-curve cryptography. `https://www.youtube.com/watch?v=dFUJmzcj0iw\#t=59m50s`. Online: Accessed 5 Nov 2015.

[51] Parsia Hakiman. Tales from the crypto leaking aes keys. `http://parsiya.net/blog/2015-01-06-tales-from-the-crypt-o-leaking-aes-keys/`. Online: Accessed 23 Sept 2015.

[52] Rob Heaton. The padding oracle attack - why crypto is terrifying. `http://robertheaton.com/2013/07/29/padding-oracle-attack/`, 2013. Online: Accessed 30 Oct 2015.

[53] Martin Hell. Cryptography lecture 2-3. *Lunds Tekniska Högskola*, 2015. Accessed 30 Oct 2015.

[54] Scott Helme. Perfect forward secrecy - an introduction. `https://scotthelme.co.uk/perfect-forward-secrecy/`, 2014. Online: Accessed 1 Nov 2015.

[55] Radu Igret. Wiring guidelines for rs-485 networks. *BobTech embedded controls*, 2011. Accessed 30 Nov 2015.

[56] RESmith Inc. Rs485 - frequently asked questions. `http://www.rs485.com/pfaq.html`. Online: Accessed 30 Nov 2015.

[57] Avi Kak. Lecture 13: Certificates, digital signatures, and the diffie-hellman key exchange algorithm. *Purdue University, West Lafayette, USA*, 2015. Accessed 14 Nov 2015.

[58] San Jose California USA Keri Systems, Inc. Solutions white paper – wiegand interface readers have 4 core problems. Accessed 23 Nov 2015.

[59] Maria Kihl. *Datakommunikation en inledande översikt.* Studentlitteratur AB, 2 edition, 1996.

[60] Richard D. Kuhn, Vincent C. Hu, W. Timothy Polk, and Shu-Jen Chang. Introduction to public key technology and the federal pki infrastructure. *National Institute of Standards and Technology*, 2001. Accessed 12 Nov 2015.

[61] Kvaser. What is galvanic isolation? why do i need it? `http://www.kvaser.com/faq/what-is-galvanic-isolation-why-do-i-need-it/`. Online: Accessed 30 Nov 2015.

[62] RSA Laboratories. 3.1.5 how large a key should be used in the rsa cryptosystem? `http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/how-large-a-key-should-be-used.htm`. Online: Accessed 19 Oct 2015.

[63] IP Location. What is a tcp/ip? `https://www.iplocation.net/tcp-ip`. Online: Accessed 30 Nov 2015.

[64] luotuofushi.net. What are disadvantages of tcp/ip. `http://www.luotuofushi.net/what-are-disadvantages-of-tcp-ip-84085261/`, 2015. Online: Accessed 5 Nov 2015.

[65] Zack Martin. Oak ridge national labs deploys combination piv, civ smart card ecosystem. 2014. Accessed 1 Nov 2015.

[66] NearFieldCommunication.org. How nfc works. `http://www.nearfieldcommunication.org/how-it-works.html`. Online: Accessed 12 Nov 2015.

[67] National Institute of Health. The personal identity verification (piv) process. `http://www.ors.od.nih.gov/ser/dpsac/Training/Pages/video.aspx`, 2013. Online: Accessed 15 Oct 2015.

[68] Australian Government Department of Human Services. Plaid application specification. Accessed 18 Oct 2015.

[69] National Institute of Standards and Technology. Block cipher modes. `http://csrc.nist.gov/groups/ST/toolkit/BCM/index.html`, 2014. Online: Accessed 30 Oct 2015.

[70] National Institute of Standards and Techonology. About personal identity verification (piv) of federal employees and contractors. `http://csrc.nist.gov/groups/SNS/piv/`, 2014. Online: Accessed 3 Dec 2015.

[71] National Institute of Standards and Techonology. Nist special publication 800-73-4. 2015. Accessed 3 Dec 2015.

[72] National Institute of Standards and Techonology. Nist special publications. `http://csrc.nist.gov/publications/PubsSPs.html`, 2015. Online: Accessed 3 Dec 2015.

[73] OpenCourseOnline. Youtube: 3 - 2 - the data encryption standard - cryptography-professor dan boneh. `https://www.youtube.com/watch?v=UgFoqxKY7cY`. Online: Accessed 3 Nov 2015.

[74] David Oswald and Christof Paar. Breaking mifare desfire mf3icd40: Power analysis and templates in the real world. 2011. Accessed 27 Nov 2015.

[75] Nicole Perlroth, Jeff Larson, and Scott Shane. N.s.a. able to foil basic safeguards of privacy on web. `http://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html?hp\&pagewanted=all\&\_r=0`, 2013. Online: Accessed 1 Nov 2015.

[76] Robert Pierce. Youtube: Elliptic curve diffie hellman. `https://www.youtube.com/watch?v=F3zzNa42-tQ`. Online: Accessed 28 Oct 2015.

[77] The Linux Information Project. Tcp/ip definition. `http://www.linfo.org/tcp\_ip.html`, 2005. Online: Accessed 2 Dec 2015.

[78] Nasdaq Press Release. gemalto receives 2015 digital payments award from juniper research. 2015. Accessed 1 Nov 2015.

[79] R.L. Rivest, A. Shamir, and L.M. Adleman. Cryptographic communications system and method. September 20 1983. US Patent 4,405,829.

[80] Robotshop. Wiegand data format. Technical report. Accessed 28 Nov 2015.

[81] IDmachines LLC Salvatore D'Agostino, CSCIP. The piv alphabet soup: Piv, piv-i, piv-c, civ-c, idm. `http://www.fips201.com/news-item/the-piv-alphabet-soup-piv-piv-i-piv-c-civ-c-idm/`, 2011. Online: Accessed 18 Oct 2015.

[82] Bruce Schneider. The nsa is breaking most encryption on the internet. `https://www.schneier.com/blog/archives/2013/09/the\_nsa\_is\_brea.html\#c1675929`, 2013. Online: Accessed 1 Nov 2015.

[83] ITU-T Telecommunication Standardization Sector. Itu-t recommendations. `http://www.itu.int/itu-t/recommendations/rec.aspx?rec=X.509`, 2015. Online: Accessed 14 Nov 2015.

[84] NXP Semiconductors. Mifare - the leading brand of contactless ic products. `https://www.mifare.net/en/about-mifare/`, 2015. Online: Accessed 22 Nov 2015.

[85] NXP Semiconductors. Mifare classic family. `https://www.mifare.net/en/products/chip-card-ics/mifare-classic/`, 2015. Online: Accessed 27 Nov 2015.

[86] NXP Semiconductors. Mifare desfire ev1 4k: Mifare desfire ev1 contactless multi-application ic. `http://www.nxp.com/products/identification\_and\_security/smart\_card\_ics/mifare\_smart\_card\_ics/mifare\_desfire/series/MIFARE\_DESFIRE\_EV1\_4K.html`, 2015. Online: Accessed 27 Nov 2015.

[87] C. E. Shannon. Communication theory of secrecy systems. http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf. Online: Accessed 3 Nov 2015.

[88] Nick Sullivan. Ecdsa: The digital signature algorithm of a better internet. `https://blog.cloudflare.com/ecdsa-the-digital-signature-algorithm-of-a-better-internet/`, 2014. Online: Accessed 3 Nov 2015.

[89] S.W.Smith. Hardware security modules. 2010. Accessed 16 Nov 2015.

[90] Introduction to Cryptography by Christof Paar. Youtube: Lecture 8: Encryption standard (aes) by christof paar. `https://www.youtube.com/watch?v=NHuibtoL\_qk`. Online: Accessed 1 Nov 2015.

[91] Luca Trevisan. Notes for lecture 1. *U.C Berkeley*, 2009. Accessed 25 Oct 2015.

[92] George W.Bush. Homeland security presidential directive/hspd-12. *Office of the Press Secretary, White House*, 2004. Accessed 3 Dec 2015.

[93] Wikipedia. Ipv4 header format. `https://en.wikipedia.org/wiki/IPv4`, 2015. Online: Accessed 3 Dec 2015.

[94] YongBin Zhou and DengGuo Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing, China*, 2005. Accessed 6 Nov 2015.

[95] Çetin Kaya Koç. Advanced encryption standard. `http://cs.ucsb.edu/~koc/ns/docs/slides/05-aes.pdf`. Online: Accessed 3 Nov 2015.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS

Date of issue
February 2016

Document Number
ISRN LUTFD2/TFRT--6003--SE

Author(s)
Michael Kapusta
Nicklas Lindstrom

Supervisor
Mathias Bruce, Axis Communications
Johan Adolfsson, Axis Communications
Martina Maggio, Dept. of Automatic Control, Lund University, Sweden
Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle
Access Control With High Security Credentials

Abstract

Developing security regardless of its format is a constant cat and mouse game were adversaries are either in the midst of trying to crack your solution, or they may have already cracked it. A cryptographic algorithm may be unfeasible to crack from a mathematical perspective but as long as a human being is the one developing the solution, a human error is always possible.

A large quantity of the current security solutions on the Physical Access Control Systems market are, as will be shown in this thesis, riddled with human errors. Security systems that are portrayed by their developers as secure even though they are not, give the users a false sense of security. The insecure Physical Access Control Systems are, as will be shown in this thesis, most frequently a result of proprietary solutions by the developers.

The thesis analyses and evaluates various authentication and authorization techniques with a high level of security for smart cards and smartphones, within the scope of Physical Access Control Systems. This includes an analysis of standards and protocols such as PIV, PLAID, FICAM and FIPS 201 with respect to their cryptographic properties, workflows and user management. The thesis also includes prototyping of such functionality on an embedded system in combination with a smartphone.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-68

Recipient's notes

Security classification

http://www.control.lth.se/publications/