

# Finding Vulnerabilities in Low-Level Protocols

Nordine Saadouni

## **Abstract**

THIS IS ALL COPIED ... I MUST CREATE MY OWN ABSTRACT!!!  
In this paper we describe attacks on PKCS#11 devices that we successfully mounted by interacting with the low-level APDU protocol, used to communicate with the device. They exploit proprietary implementation weaknesses which allow attackers to bypass the security enforced at the PKCS#11 level. Some of the attacks leak, as cleartext, sensitive cryptographic keys in devices that were previously considered secure. We present a new threat model for the PKCS#11 middleware and we discuss the new attacks with respect to various attackers and application configurations. All the attacks presented in this paper have been timely reported to manufacturers following a responsible disclosure process.

Smart cards are used commercially and within industry for authentication, encryption, decryption, signing and verifying data. This paper aims to look into how the smart card interacts with an application at the lower level. PKCS#11 (public key cryptography system?) is the standard that is implemented at the higher level and then broken down into command/response pairs sent as APDU traffic to and from the smart card. It is the APDU low-level protocol that will be analysed to see if any vulnerabilities are present with regard to the smart cards tested.

Once I have completed the paper it is better to do similar to what is done above and write a brief summary of the entire paper!

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Cryptoki . . . . .	3
2.2	ISO 7816 . . . . .	3
2.3	Tools developed and used . . . . .	3
<b>3</b>	<b>Related Work</b>	<b>4</b>
<b>4</b>	<b>Methodology - Technical Aspects</b>	<b>5</b>
4.1	C Login - Authentication Procedure . . . . .	5

# Chapter 1

## Introduction

Smart-cards are formally known as integrated circuit cards (ICC), and are universally thought to be secure, tamper-resistant devices. They store and process, cryptographic keys, authentication and user sensitive data. They are utilised to perform operations where confidentiality, data integrity and authentication are key to the security of a system.

Smart-cards offer what seems to be more secure methods for using cryptographic operations. This is partly due to the fact that the majority of modern smart-cards have their own on-board micro-controller, to allow all of these operations to take place on the card itself, with keys that are unknown to the outside world and stored securely on the device. Meaning the only actor that should be able to perform such operations would need to be in possession of the smart-card and the PIN/password. In many industries, for applications such as:

- Banking/ payment
- E-commerce
- Sim cards/ telecommunications
- Healthcare
- Public transport

smart-cards are used due to the security they are believed to provide.

The most common API (application programming interface) that is used to communicate with smart-cards is the RSA defined PKCS#11 (Public Key Cryptography Standard). Also known as 'Cryptoki' (cryptographic token interface, pronounced as 'crypto-key'). The standard defines a platform-independent API to smart-cards and hardware security modules (HSM). PKCS#11 originated from RSA security, but has since been placed into the hands of OASIS PKCS#11 Technical Committee to continue its work (since 2013). [reference wikipedia PKCS#11].

In the previous X years, literature has shown a great deal of research has examined the PKCS#11 API.

## Chapter 2

# Background

### 2.1 Cryptoki

PKCS#11 is all written in the C programming language, and the standard only includes the header files, in which card vendors all must conform too as provide the platform independence it wished to achieve. This allows proprietary implementations by card vendors, while still allowing a generic format for applications to utilise/ call the methods of PKCS#11.

It also consists of a 'manual' / set of instructions that must be followed in order to maintain security attributes of the API. \*\* (This will probably need to be altered) \*\*

### 2.2 ISO 7816

### 2.3 Tools developed and used

Here mention:

MiTM attack (frank someone), creates virtual smartcard (that connects to middleware), and we relay the communication to get response from smartcard  
Opensc PKCS11-tool pcsd (for sniffing traces)

## Chapter 3

# Related Work

literature review

## Chapter 4

# Methodology - Technical Aspects

### 4.1 C Login - Authentication Procedure

Initially we used the 'pcscd' program, that was part of the PC/SC software package to sniff all APDU communication between smart-card and the computer.

We found that when the C\_login function is called, it is broken down into the following APDU communication command/ response pairs.  
list them (maybe picture)

# Bibliography

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.
- [3] Knuth: Computers and Typesetting,  
<http://www-cs-faculty.stanford.edu/~uno/abcde.html>