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 analysied 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

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 preform 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 person that should be able to preform such operations would need to be in possession of the 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. [maybe change this last section]

The most common API (application programming interface) that is used to communicate with smart-cards from applications/ computers 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 cryptographic tokens such as smart-cards and hardware security modules (HSM). 'Cryptoki' 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].

'Cryptoki' 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 maintin security attributes of the API.

Still to complete in this chapter:
Briefly mention ISO 7816-4 standards, and how it links to cryptoki (more detailed explanation will be given in the 'background' chapter)
Then move onto mention that in the past decade or so attention has mainly been on the PKCS#11 standard, with little attention paid to the underlying application data protocol. (the related work section will list such security scrutinies of the API and attacks found)

Before moving on explain the purpose of the paper. To search for vulnerabilities within the lower-level communication, that would contradict PKCS#11 standard. (mention some fundamental basics of the standard [never reveal keys, marked 'sensitive', in an un-encrypted communication stream])

Chapter 2

Background

Probably better to give a briefer explanation in the introduction and then explain it more in depth in this chapter or CRYPTOKI and ISO7816

2.1 Cryptoki

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 pcscd (for sniffing traces)

Chapter 3

Related Work

literature review

Chapter 4

What I have done so far (this needs a better name!)

4.1 C Login

Explain authentication

Bibliography

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX 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`