P20

A *Hardware Security* Module for the Raspberry Pi

At the end of this laboratory, you should have successfully built an I; Matto-based *hardware security module* attached via SPI to the Raspberry Pi.

Preparation time : 9 hours

Lab time : 9 hours

Items provided

Tools : None

Components : None

Equipment : DVI-D capable monitor

Software : Win32 disk imager

Qt5 Raspbian SD images

Items to bring

- Essentials. A full list is available on the Laboratory website at https://secure.ecs.soton.ac.uk/notes/ellabs/databook/essentials/
- Raspberry Pi boot image
- Raspberry Pi
- Raspberry Pi power supply
- Raspberry Pi SD card
- Raspberry Pi HDMI to DVI-D cable
- Raspberry Pi keyboard
- Raspberry Pi mouse
- Il Matto board

Academic Integrity – *If you undertake the work jointly with other students, it is important that you acknowledge this fact in your logbook. Similarly, you may want to use sources from the internet or books to help answer some of the questions. Again, record any sources in your logbook.*

Aims, Learning Outcomes and Outline

This laboratory exercise aims to:

- Develop confidence in the use of SPI communications.
- Develop an understanding of handshakes and synchronisation.
- Develop the ability to construct a working cryptosystem.

Having successfully completed the lab, you will be able to:

- Construct a working two-processor system.
- Make effective decisions about communications protocols.

1 Background

You are asked to set up a file encryption system for the Raspberry Pi which uses an Il Matto as a *hardware security module*. The HSM will hold a long-term secret key which must never leave the Il Matto. When a file is to be encrypted or decrypted, an additional session (file) key is passed from the Raspberry Pi to the Il Matto and is mixed with the long-term key. This mixed key is used to initialise an RC4 pseudo-random number generator (PRNG) in the Il Matto which issues a stream of bytes back to the Raspberry Pi. The file for en- or de-cryption (the operations are the same) is then XORed byte-by-byte with the RC4 stream.

2 Communication

The byte transfers, both of session key and PRNG stream should be over SPI; you should set up the Raspberry Pi as master, choosing an appropriate speed, and the Il Matto as slave.

The master will need to initiate transfers only when the slave is ready; we suggest you connect an additional two GPIO pins between Raspberry Pi and Il Matto to serve as request and acknowledge handshakes.

3 Preparation

You will need to find copies of the RC4 and RC4 key-scheduling algorithms, and ideally test them on the Raspberry Pi.

You will need to research GPIO and SPI interfaces on the Il Matto and Raspberry Pi; we believe all necessary modules are present in the Raspbian kernel but will need to be enabled and the protections adjusted.

4 Required Programs

You will produce C or C++ code for both the Raspberry Pi and the Il Matto. The user program on the Raspberry Pi should take a single argument, the session key, and should en- (de-)crypt standard input to standard output. This should work reliably, without any artificial time delays introduced into either program.

5 Optional Additional Work

Marks will only be awarded for this section if you have already completed all of Section 3 to an excellent standard and with excellent understanding.

1. Modify the Raspberry Pi program to prompt for a password which is *not* echoed to the screen.

2. Lock the Il Matto to prevent extraction of the secret key.

3. Create a graphical interface for the Raspberry Pi program.

4. Research the vulnerabilities of RC4 and modify your key mixing and key scheduling algorithms to improve security.

Revision History

08 April 2014	Denis Nicole (dan)	2014 version of this lab created