RFID++:

Testing existing security of RFID cards for next-gen authentication systems.

Saket Upadhyay 18BCY10079

Saket Upadhyay

Bachellor of Technology (2022),
Department of Cybersecurity and Digital Forensics,
School of Computer Science and Engineering,
Vellore Institute of Technology, Bhopal

Capstone Project

RFID++ Testing existing security of RFID cards for next-gen authentication systems.

> Team 31

Project Outcomes

> Creating a Code-Base for executing RFID based authentication module in Raspberry-Pi and Raspberry-Pico using RFID-RC522 chipset.

> Creating an open-source code base for the Card Clone Demo for academia to use in security lectures and study of the basic low-level working of RFID-based authentication systems.

> Shed some light on the existing research on RFID authentication.

> An attempt to summarize and find potential for future work in this field.

01

Project Introduction

What is RFID, its properties, issues and security.

What is RFID?

RFID (Radio Frequency Identification) is a crucial application technology in the Internet of Things (IoT) technology. The terminal device needs to be authenticated before accessing the IoT network to avoid security holes. Due to the limited resources of the tag side in passive RFID systems, ultra-lightweight RFID authentication protocols are often used in such systems.

What is RFID?

Apart from privacy and security issues, the computation capability and memory of the tags are very limited as compared to the backend server and the reader;
Therefore, an RFID-based authentication protocol suffers from high computational overhead.

Benefits of RFID

- Small size
- Light weight
- Affordability
- Long shelf life (up to 20+ years)

Motivation

The seamless integration and ease of use of RFID systems inherit some fundamental security issues.
The limited computing power of silicon used in RFID tags and cards prohibits the use of more complex cryptographic functions to encrypt the data. Manufacturers usually end up using standard encoding to store the data which is not enough for security applications nowadays as mobile computing devices are getting more powerful and can intercept and process the data transmitted and manipulate them while having a small footprint, this introduces a big risk for enterprises and end-users in terms of secure implementation of such devices.

Problem Statement

With an exponential increase in electronics in our day-to-day life, our bias is increasing toward contactless/seamless authentication systems as they are easy to use and maintain. This is introducing more and more RFID-based technology into the consumer market, most of them are not properly regulated and end up as off-brand door locks or security systems on popular e-commerce websites like Amazon and Alibaba.

They create an illusion of security but, are pretty easy to override.

This creates a serious problem, and the solution is the need to create a robust and affordable security algorithm/strategy for consumer-level RFID security and authentication.

Project's Expectations

This project explores the existing and recently proposed RFID protocols and strategies for authentication systems and attempts to find the research gap in the field. Also, we aim to create a demonstration of the RFID Card Cloning procedure where we will try to clone an existing production RFID ID card.

02

Hardware and Software Requirements

List of components / Modules

Hardware

a) Microcontroller (RP2040)
b) Microprocessor (Raspberry Pi Zero W 1GHz, single-core CPU)
c) A Laptop/Desktop Computer
d) Bread Board
e) RC522 RFID Card Reader Module 13.56MHz
f) Jumper Cables
g) Read/Write RFID Blank Card
h) Read/Write RFID Blank Token
i) A working real-world RFID Card/Token
j) Male to Female Connector Headers
k) Soldering Equipment

Software

We don’t need any specific proprietary software/stack but we will need some open-source frameworks and programming language support, as follows-

a) Linux Operating System for Microprocessor.
b) Python3.
c) C/C++ development toolchain.
d) MicroPython toolchain for RP2040.
e) Opensource RFID libraries.

03

Setting up USB Serial Communication

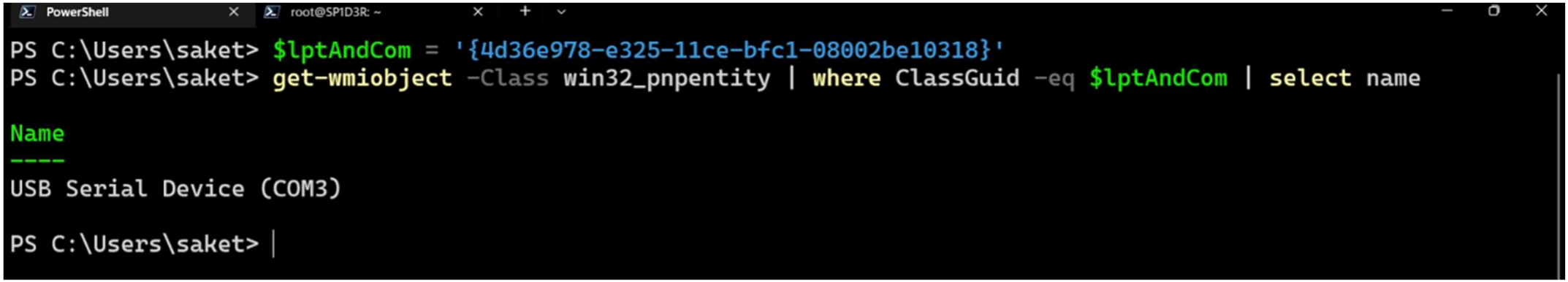
Establishing a serial communication channel via minicom to communicate with MicroPython instance

Hardware Configuration

RC522 RFID Reader Module	Raspberry Pi Pico
VCC	3.3V
RST	GP0
GND	GND
IRQ	Not connected
MISO	GP4
MOSI	GP3
SCK	GP2
SDA	GP1

Search for allocated COM port

We can search for allocated COMφ port (COM3 in this case) on windows via following commands in PowerShell-
$(\varphi \in N)$



Connect via minicom on WSL

After finding the port, we can activate WSL in Windows and use minicom to connect to the device's serial interface via-

```
minicom -S /dev/ttySφ
```

Where φ is the COM port in COMφ, where $\varphi \in N$.

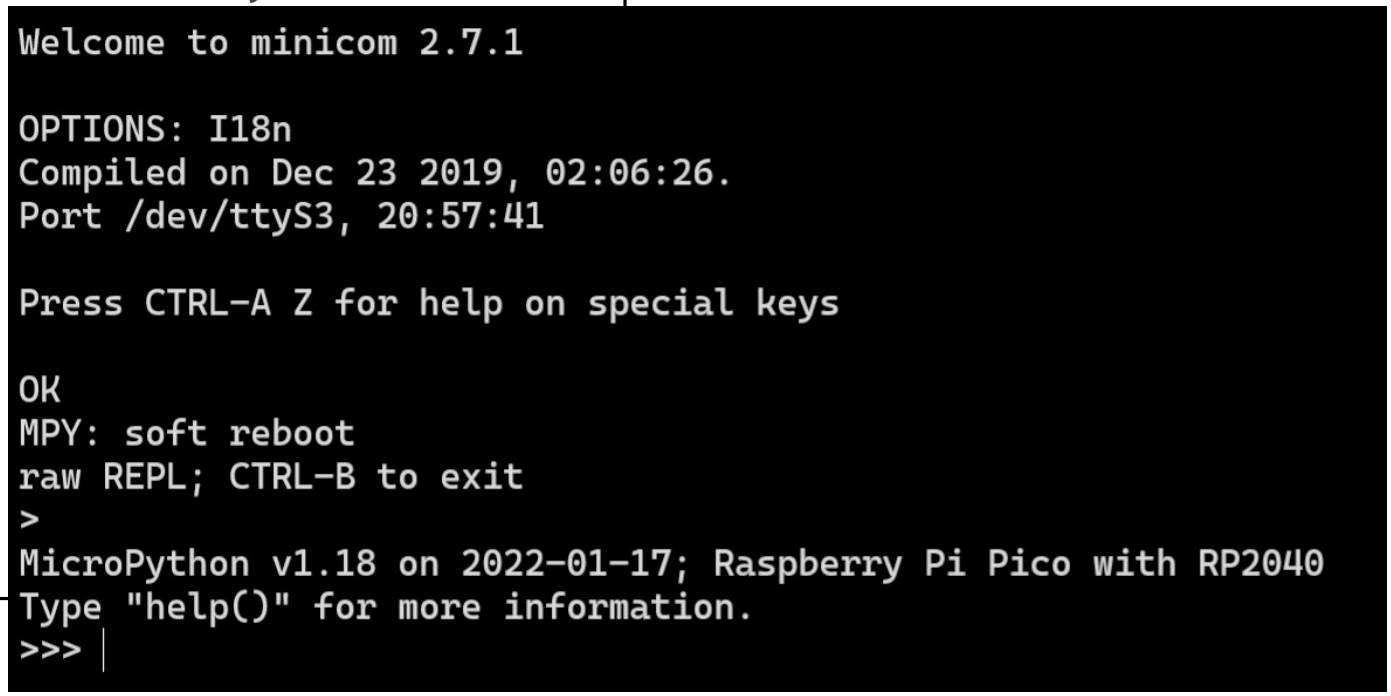
Connect via minicom on WSL

> CTRL+D in minicom interface will force Pico to reboot.

> CTRL+B then will show MicroPython interface.

> CTRL+A+C will clear the screen.

> CTRL+A+X will exit minicom.



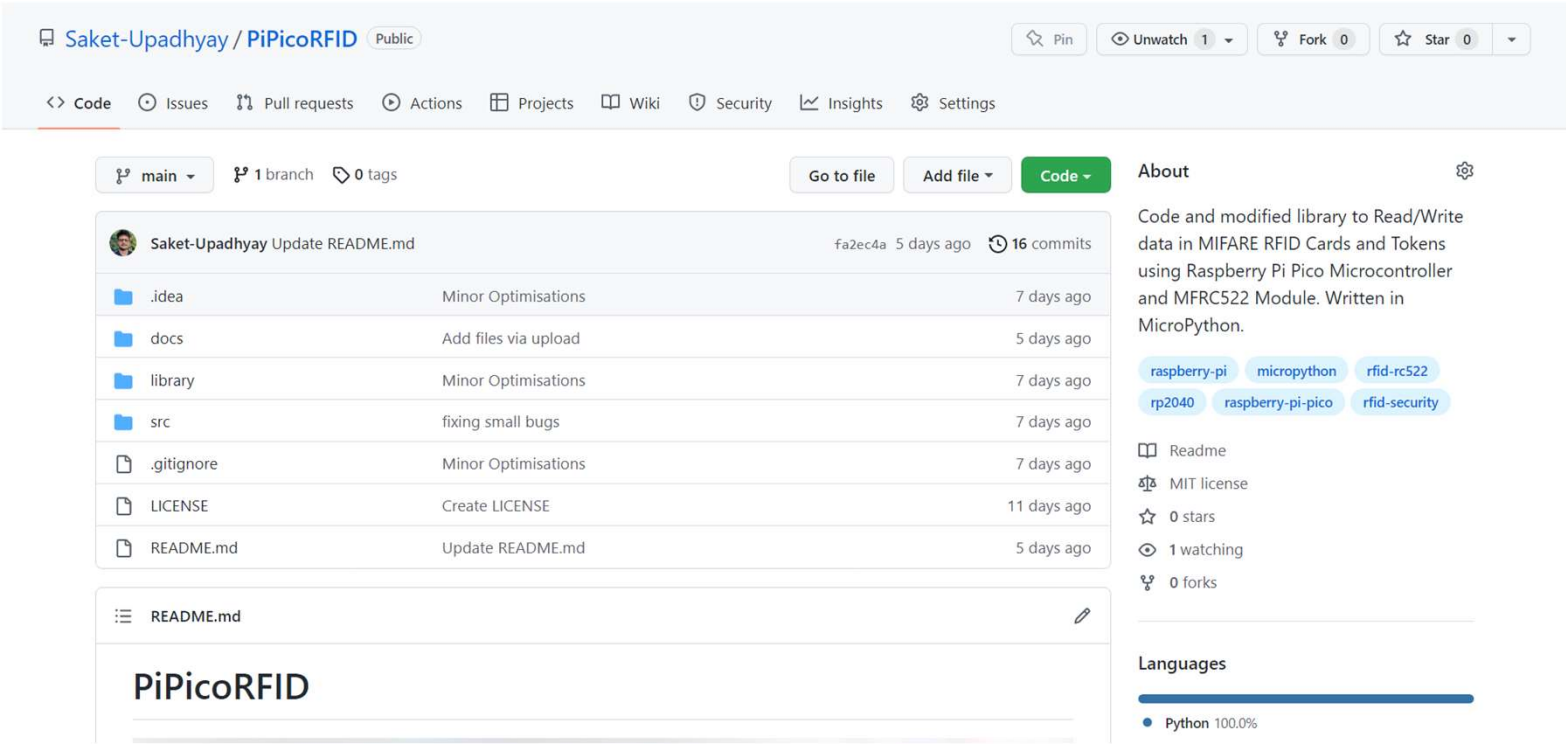
04

Cloning Card Data

Running the scripts from micropython interface

Opensource Code

All the code generated during this project is open-sourced under MIT license (See APPENDIX I) @ https://github.com/Saket-Upadhyay/PiPicoRFID



Read/Write MIFARE Card

The readRFID.py and writeRFID.py scripts will read from and write into the card respectively.

The program can read/write 16 bytes of data from the card at address 0x08.

The KeyGen.py generates random 16byte data consisting of ASCII printable small and capital case letters and numbers, then stores it in a byte array that can be used to write the data in the card.

Read/Write MIFARE Card

```
>>> import writeRFID
>>> writeRFID.RUN()
Init. rp2

Place card before reader. WRITE ADDR: 0x08

CARD DETECTED
 -  TAG TYPE : 0x10
 -  UID      : 0xfccee838

DATA WRITTEN TO ADDRESS 0x08
EXITING PROGRAM
>>> readRFID.RUN()
Initialising Module=> rp2

Place card before reader. READ ARRD: 0x08

CARD DETECTED
 -  TAG TYPE : 0x10
 -  UID      : 0xfccee838

DATA: ABCDABCDABCDABCD
RAW DATA: ['0x41', '0x42', '0x43', '0x44', '0x41', '0x42', '0x43', '0x44', '0x41',
```

Experiment

We can import SampleRFIDScanner.py and CloneCardData.py to run the experiment.

Execute the SimpleRFIDSamle.RUN() and bring the original card and it should result in access granted, run it again, and scan other tokens that should result in Access Denied message from the program.

To clone the data, import CloneCardData and call the RUN() function. Follow the instructions and the data from the original card will be copied to the second blank card/token.

Run the SampleRFIDScanner.RUN() and the clone token should be accepted as original.

Clone Card Data

```
>>> import CloneCardData
>>> CloneCardData.RUN()
Initialising Module=> rp2

Place Original card before reader.

CARD DETECTED
 -  TAG TYPE : 0x10
 -  UID      : 0xd4726a69

[68, 119, 77, 111, 48, 71, 56, 73, 109, 109, 85, 76, 115, 74, 68, 101]
b'DwMo0G8ImmULsJDe'
remove the card.

Place Clone card before reader.

CARD DETECTED
 -  TAG TYPE : 0x10
 -  UID      : 0x238b4715

DATA WRITTEN TO ADDRESS 0x08
>>>
```

05

Results and Conclusion

Experiemnt results and conclusion of the study.

Results

After the cloning process, the second RFID card should have the same data as in the Original Card. That should enable us to get “Access Granted” using clone card from the SampleRFIDScanner.RUN()

Limits of this project

The limitation of this work is that we cannot change the UID of the RFID card with available hardware.

Generally, the UID of a general MIFARE 1K Card is the first 4 bytes in block 0 of sectors 0 and it cannot be modified after being manufactured. But the 1st generation of UID Changeable Card (named Chinese Magic Card) can be changed by an external device, such as PN532, ACR122U, PM3, etc. The device needs to be used with nfc-mfsetuid on Linux.

Potential Future Work

Lightweight and ultra-lightweight protocols are considered the most suitable for the current applications. Another vital aspect when considering the appropriate RFID protocol is the security resistance to the attacks.

Researchers are encouraged to pay attention to the forward and backward compatible security since most protocols do not reflect on these two types of attacks. Finally, maintaining the basic security requirements for an RFID system is required to achieve protection against major attacks.

Thank You

APPENDIX I (MIT License)

MIT License

Copyright (c) 2022 Saket Upadhyay

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.