RFID Project

Computer System Security

Mateo Fortea Dugo & Lucía Cabezuelo Pérez

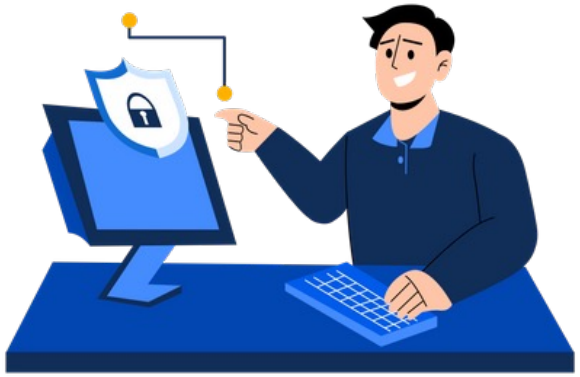
Professor Miltos Grammatikakis

Department of Electrical & Computer Engineering

23th January 2024

Index

1 Introduction

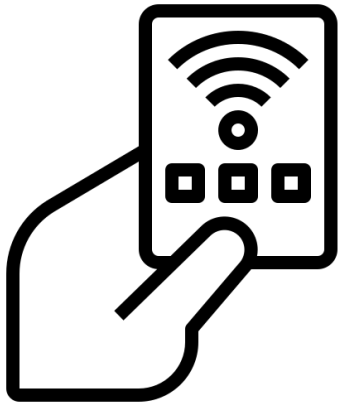


Cybersecurity

Become more important nowadays

Can be used in a lot of fields

RFID tecnhology is vulnerable by itself



RFID Technology

RFID = Radio-Frequency Identification

Applications: inventory tracking, access control, logistics ...

1 Introduction

RFID Security Challenges

Cloning vulnerability
Signal interception
Replay attacks

The Project goal

A RFID security system enhanced with encryption to address and mitigate these vulnerabilities.

2 Motivation

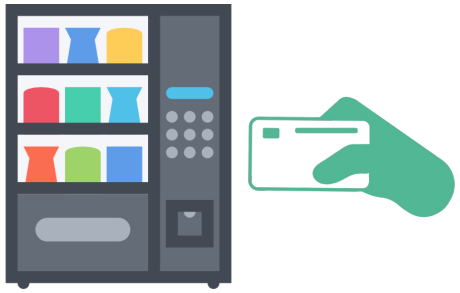
PROTECTION OF SENSITIVE DATA

LIMITATIONS OF CURRENT SYSTEMS

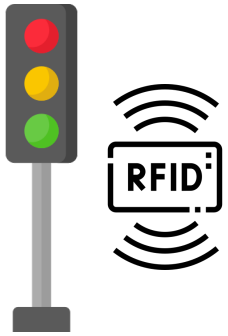
CONTRIBUTION TO THE COMMUNITY

3 Related work

RFID RAILWAY SIMULATION



VENDING SIMULATION APP

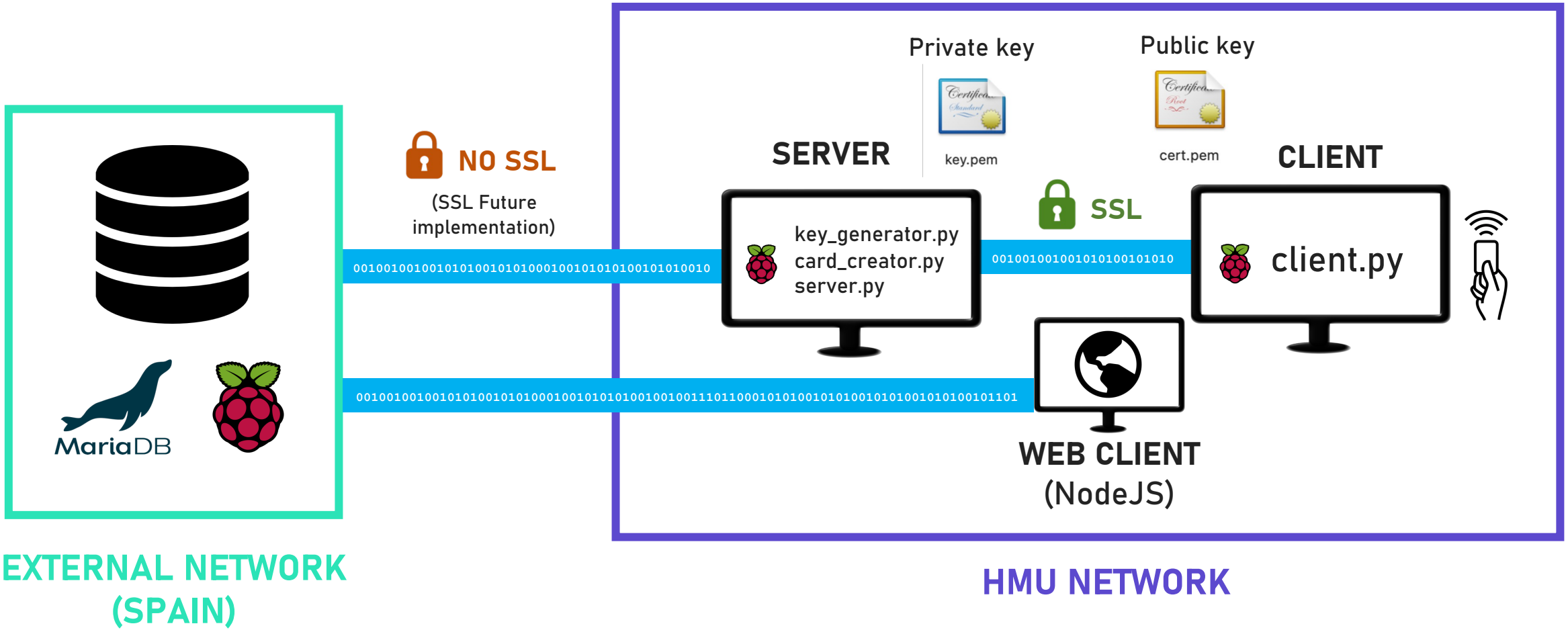


RFID TRAFIC LIGHTS

- Uses writer/reader RFID
- Uses LabView (Visual coding) instead of Python
- Uses AES encryption for the data

**Refered in the Bibliography*

4 Methodology / Physical structure



4 Methodology / Used technologies

RFID

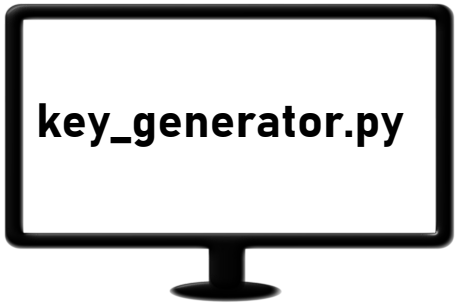
AES-256, FERNET & PBKDF2

WEBSOCKET OVER SSL

MariaDB

MARIADB DATABASE

4 Methodology / Functioning



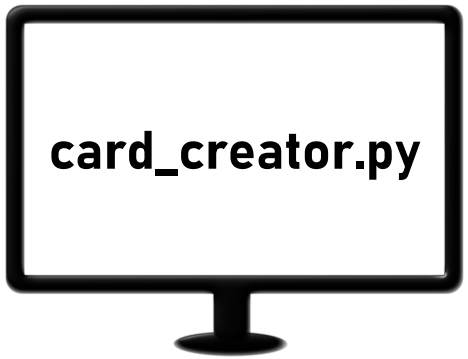
1 Input new FERNET PIN

Creates new AES "Master Key" 2

AES-256 Key

3 Uses **FERNET** for encrypt password usages with a PIN

4 Key stored in the server as a encrypted file



PIN for unlock master key

Enter new username 5

Enterr new user password

```
def derive_diversified_key(master_key, user_id):
    combined = master_key + user_id.encode()
return hashlib.sha256(combined).digest()
```

6

Master key + **User ID** hashed combined

Diversified key (32 bytes)

4 Methodology / Functioning

***If simulation mode ON.*
Creates files card_0.txt*

User password

Diversified key (32 bytes)

Random nonce (16 bytes)

```
encrypt_aes(password, diversified_key, nonce)
```

7

User password encrypyted

Approach the cards for writing

8

4 cards
(48 bytes per card)

- **Diversified key (Card 1 + Card 2)**
- **Random nonce (Card 3)**
- **Encrypted user password (Card 4)**

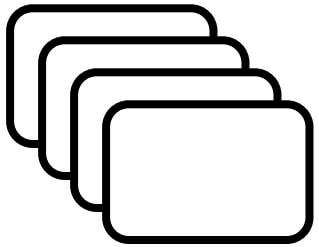
9

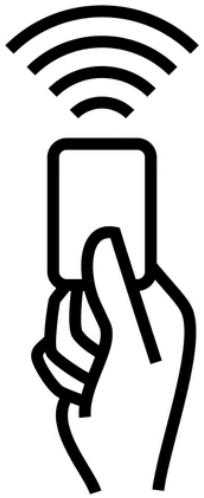
Database

- **Username**
- **Diversified key**
- **Random nonce**
- **Hashed (not encrypted) user password**

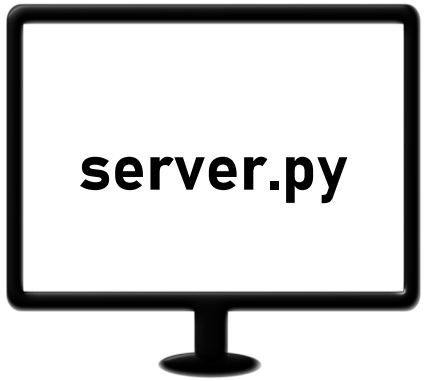
4 Methodology / Functioning

10 Input username (if exists) then approach the cards in the correct order





4 Methodology / Functioning



11 After running server, input Fernet PIN for “unlock” Master Key usage

12 Check username exists (compares input with DB stored usernames)

13 Uses Master Key for generate same derivative key

14 Compares stored nonce in the DB and card nonce → If not, refuse login in the server (Card maybe altered)

15 Decrypt the password with derivative key

16 Hash the decrypted password and compares with the also hashed password stored in the database → If it match, user log in

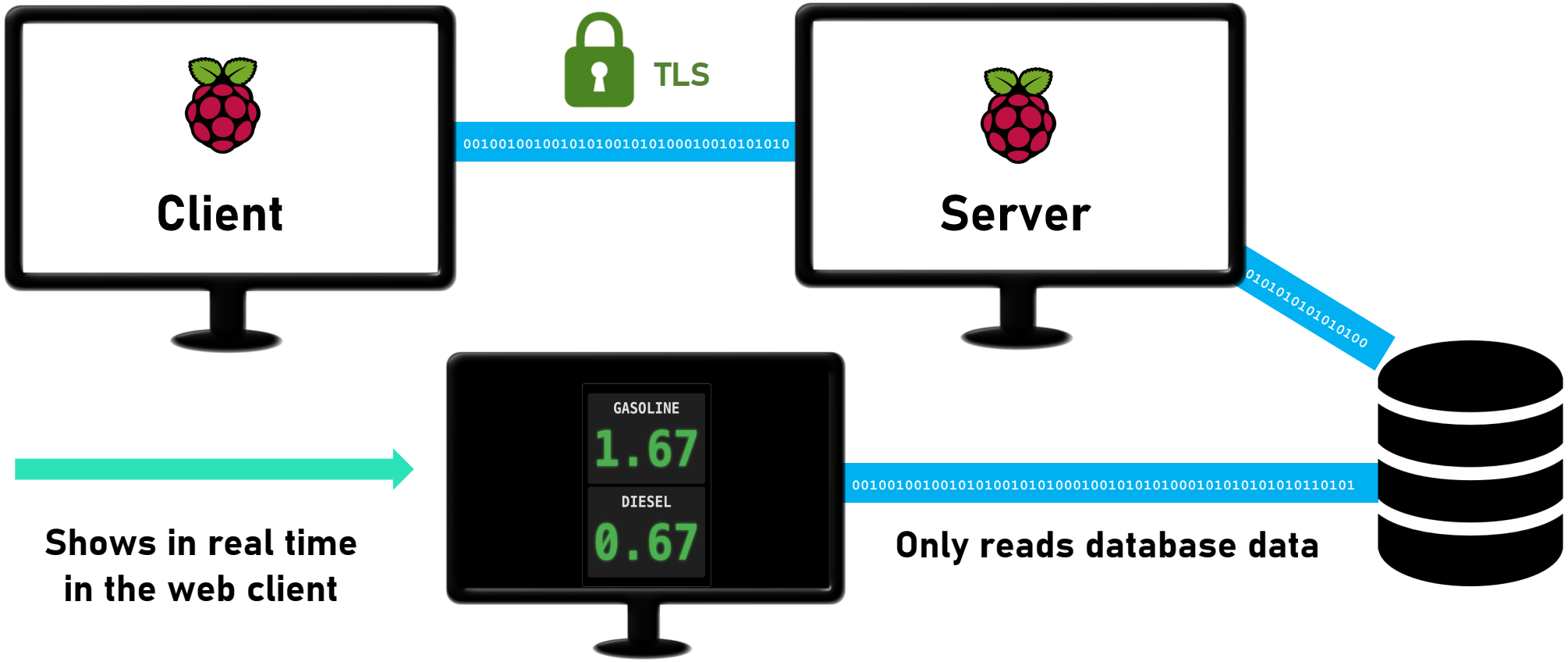
4 Methodology / Functioning

17

Shows a menu

Could changes gas station prices

18

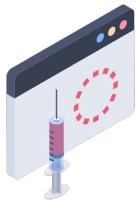


Shows in real time in the web client

Only reads database data

4 Methodology / Possible attacks







TYPE OF ATTACK	REPLAY ATTACKS	RFID CARD CLONING	SQL INJECTION	MAN-IN-THE-MIDDLE	DoS ATTACKS	MASTER KEY THEFT	ZERO DAY EXPLOITS
COVERED	NO	PARTIAL	YES	PARTIAL	NO	YES	NO
CAUSE	Everyone could use the same data in the cards for authenticate	Covered only if not all cards are stolen	Parameterized SQL queries	Partially through secure channel (SSL/TLS)	Requires network-level protection measures	The master key is encrypted too by itself	Code must be updated for avoid posible code exploits

5 Development

python

Main language used for the project

Visual Studio Code

IDE used together with Linux/UNIX terminal

GitHub

Distributed version control platform

node JS

Framework used for the web client

MariaDB

SQL Relational Database (DBMS)

5 Development / Main Libraries

python

CRYPTOGRAPHY
Provides cryptographic tools for secure encryption and decryption operations.

DOTENV
Manages and loads environment variables from a .env file

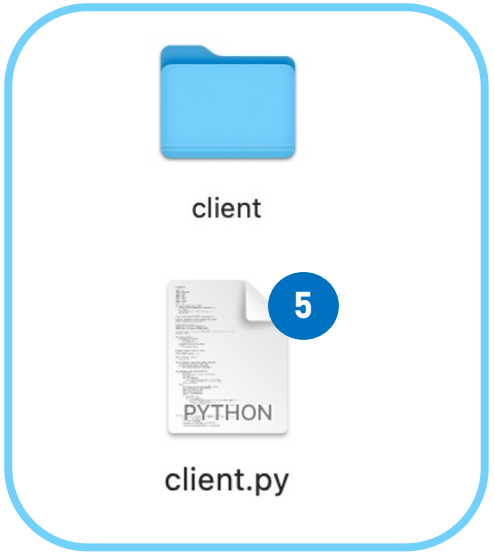
MARIADB
Enables database interactions with a MariaDB server

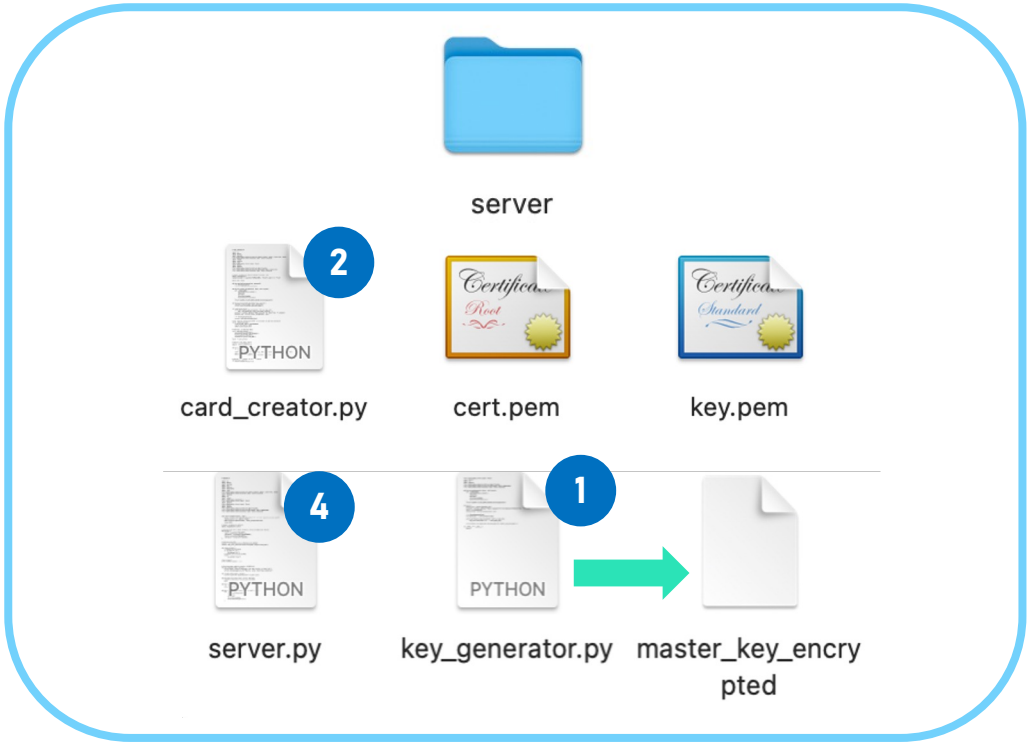
WEBSOCKETS
Facilitates real-time, bidirectional communication between server and client over the web.

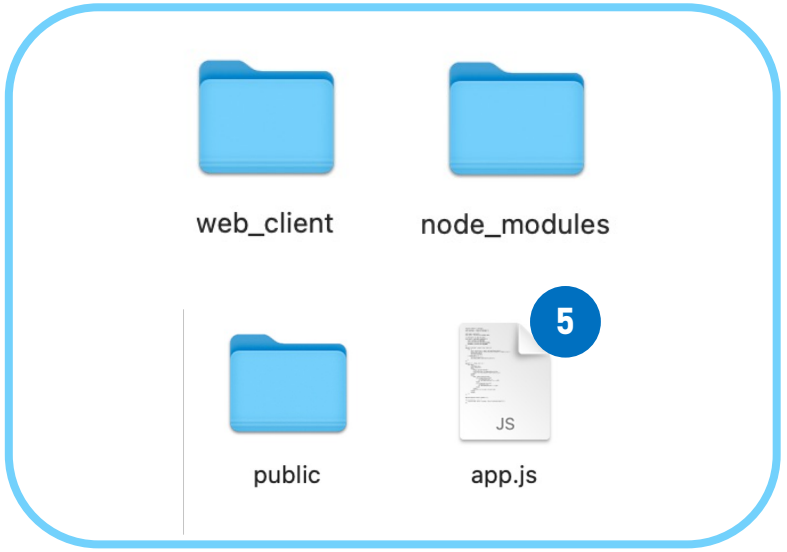
MFRC522
Interacts with RFID-RC522 readers for reading and writing RFID tags

RPi.GPIO
Controls GPIO pins on a Raspberry Pi

5 Development / Code organization





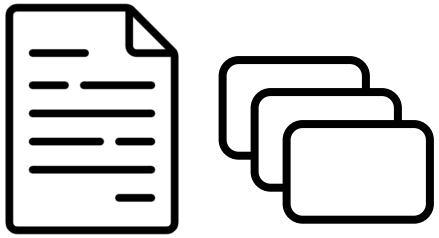


** Numbers represents logical order of execution*

cert.pem -> SSL Public Key → `openssl genpkey -algorithm RSA -out key.pem`

key.pem -> SSL Private Key → `openssl req -new -key key.pem -x509 -days 365 -out cert.pem`

5 Development / Main functions



card_creator.py

read_master_key()
Decrypts and returns the master encryption key stored in a file.

derive_diversified_key()
Generates a unique key for each user by combining the master key with the user's ID.

encrypt_aes()
Encrypts user pasword using the AES algorithm in CTR mode.

write_data()
Writes to RFID cards or simulation files.

GitHub

For view the full code:
https://gihtub.com/mfortea/RFID-PROJECT

5 Development / Main functions

client.py

read_data_from_cards()
Reads encrypted or plain data from RFID cards or simulation files.

send_card_data_to_server()
Packages and sends card data to the server for authentication.

server.py

decrypt_aes()
Decrypts data using the AES algorithm in CTR mode.

```
cipher =
Cipher(algorithms.AES(key),
modes.CTR(nonce),
```

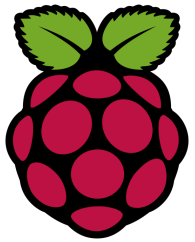
authenticate_user()
Validates the user's credentials by decrypting and comparing card data with database records.

change_price()
Updates fuel prices in the database based on user input.

GitHub

For view the full code:
https://gihtub.com/mfortea/RFID-PROJECT

6 Results

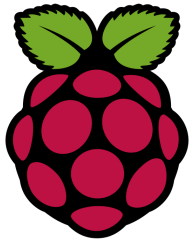


SERVER

Raspberry PI 4

Quad Core 1.8GHz CPU

8 GB RAM



CLIENT

Raspberry PI 3B +

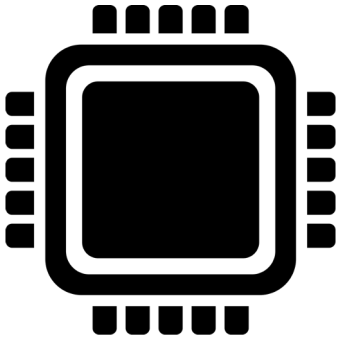
Quad Core 1.2GHz CPU

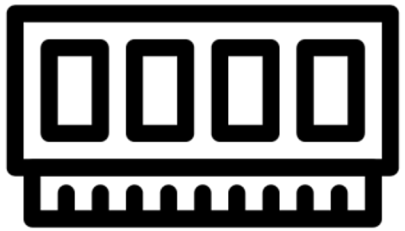
1 GB RAM

(Card creator too)

Watts meter

(For power)





Glances software

(For CPU & RAM usage)

6 Results

**Every data is average except indicated peaks*

SERVER	NO USAGE	MASTER KEY UNLOCK	SERVER RUNNING	DB LOGIN	KEY DECRYPTION	DB OPERATIONS	1 CLIENT	5 CLIENTS	10 CLIENTS
Description	Without runing any code	Decrypting M.K using fernet PIN	Server waiting for clients	Check username in the DB	AES Decryption	Updates on the DB	1 client connected	5 clients connected	10 clients connected
Power comsumption (Watts)	2,282 Watts	3,412 Watts	2,282 Watts	2,504 Watts	2,714 Watts	2,508 Watts	2,282 Watts	2,360 Watts (2,5 Watts Peak)	2,360 Watts (2,5 Watts Peak)
CPU Usage (%)	18 %	27% (Peak)	19%	20%	20%	20%	19%	20% (20,4% Peak)	20% (20,45 % Peak)
RAM Usage (MB)	990 MB	990 MB	1 GB	1 GB	1,01 GB	1,005 GB	990 MB	1,01 GB	1,01 GB (1,02 GB Peak)

6 Results

**Every data is average except indicated peaks*

CLIENT	NO USAGE	CLIENT RUNNING	1 CLIENT	5 CLIENTS	10 CLIENTS
Description	Without running any code	Client running but not connected to the server	1 client connected to the server	5 client connected to the server	10 client connected to the server
Power comsumption (Watts)	1,45 Watts	1,6 Watts	1,6 Watts	1,6 Watts 2,4 Watts (Peak)	1,6 Watts 2,4 Watts (Peak)
CPU Usage (%)	21,5 %	22 % 33 % (Peak)	22 %	22 % 23 % (Peak)	22,5 % 33 % (Peak)
RAM Usage (MB)	305 MB	310 MB	315 MB	345 MB	377 MB

6 Results

**Every data is average except indicated peaks*

CARD CREATOR	NO USAGE	DURING MASTER KEY DECRYPTION	WRITING IN THE CARDS	WRITING IN THE FILES (Simulation mode)
Description	Without running any code	Decrypting M.K using fernet PIN	Writing new data into the RFID cards	Writing the data into the files (simulates RFID cards)
Power comsumption (Watts)	1,45 Watts	2,313 Watts	2,14 Watts (Waiting for cards)	1,8 Watts
CPU Usage (%)	21,5 %	32 %	47 %	25 %
RAM Usage (MB)	305 MB	315 MB	320 MB	310

6 Results

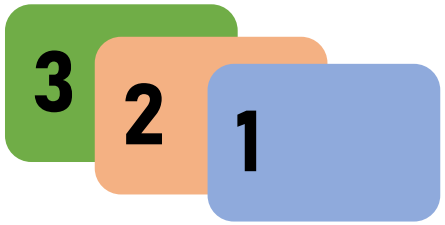
Data interpretation

- **RAM** is barely involved in the process
- **RFID** technology has an high power consumption & CPU usage
- **RAM** usage only increments when there are several clients connected
- All **decryption** process have an high power consumption & CPU usage
- Operations to the **database** increases the power consumption in the server

7 Conclusion

This project has helped us to understand everything necessary to transform an existing system into something secure and orient it towards real functionality

8 Future work



TRACKED CARDS ID

"Fast and Reliable Missing Tag Detection for Multiple-Group RFID Systems"

Exposes techniques for detect missing tags

TWO FACTOR AUTHENTICATION (2FA)

" Implementation of Two Factor Authentication based on RFID and Face Recognition using LBP Algorithm on Access Control System"

Talks about methods for implement 2FA (biometrical) together RFID technology

GPS CARD TRACKER

"Animal Situation Tracking Service Using RFID, GPS, and Sensors"

Talks about GPS located RFID devices for animals

**Refered in the Bibliography*

Bibliography

- RELATED WORK: https://upcommons.upc.edu/bitstream/handle/2099.1/19456/Mem%C3%B2ria.pdf
- https://cryptography.io/en/latest/fernet/
- https://www.tutorialspoint.com/websockets/index.htm

FUTURE WORK (Papers):

- "Fast and Reliable Missing Tag Detection for Multiple-Group RFID Systems" – https://ieeexplore.ieee.org/document/9354021

- "Implementation of Two Factor Authentication based on RFID and Face Recognition using LBP Algorithm on Access Control System" https://ieeexplore.ieee.org/abstract/document/9307564

- Animal Situation Tracking Service Using RFID, GPS, and Sensors: https://ieeexplore.ieee.org/abstract/document/5474518

9 Demo