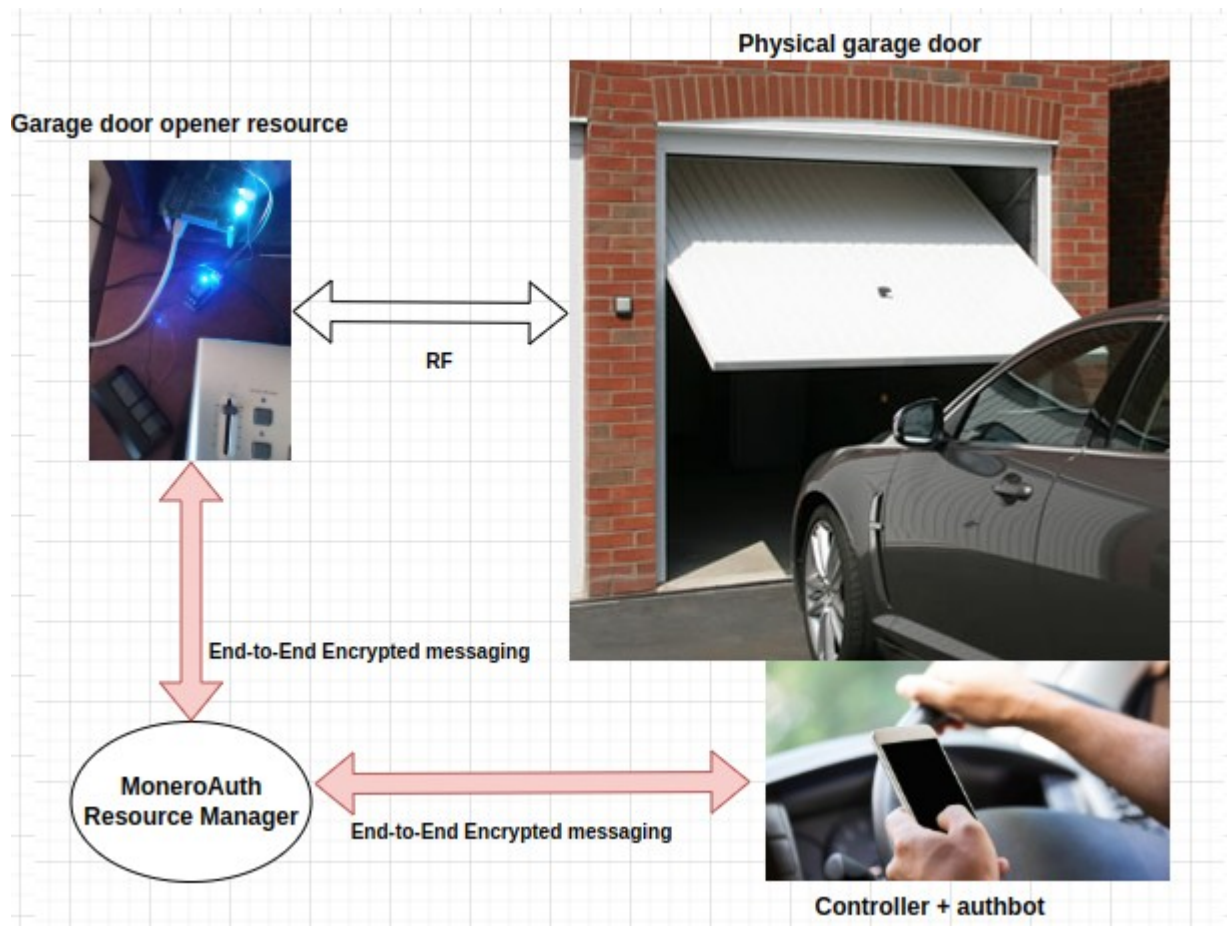


Preliminary

MoneroAuth Garage Door Opener

July 10, 2023



Description

The objective is to enable authorized individuals with mobile devices to open and close a garage door via the MoneroAuth protocol. This is accomplished by passing messages between use case actors over an end-to-end encrypted network.

The **actors** in this use case are described below:

Controller– The user who wishes to control the garage door opener via a mobile device.

Resource– The garage door opener. In this use case A Raspberry Pi (model 3 B) single-board computer running Debian version 11 (bullseye) is connected to a 5-volt relay module that controls the switch to open/close a garage door. The Raspberry Pi runs the software that controls the opening and closing of the garage door via the MoneroAuth protocol.

Resource Manager– The entity assigned to manage the garage door opener a.k.a *The Gatekeeper*. Resource Managers are responsible for authenticating and authorizing entities to access resources under its control.

Each actor has an authbot and a unique MoneroAuth digital ID used to uniquely identify the actor. An actors authbot enables the actor to communicate over the end-to-end encrypted messaging network, participate in MoneroAuth authentication/authorization activities, and automated resource management workflow.

Preconditions:

1. Controller has been authorized to open and close the garage door.
2. A QR code defining the command to activate the garage door opener has been made available to the Controller.

Postconditions:

1. The garage door has been activated (opened or closed)

Flow:

1. The Controller opens the mobile application and scans the QR code, then clicks the button in the app to activate the garage door.
2. The mobile application asks the Controller to identify the private room for the messaging and then sends the command to the Controller's authbot.
3. The Controller's authbot edits the command adding the Controller's ID and the Controller's authbot ID, invites the Resource Manager and Resource (garage door opener) to the private room (*authbots are designed to automatically accept room invites*), then sends the command to the Resource Manager.
4. The Resource Manager performs the following checks:

- Is the `resource_mgr_id` in the message equal my `resource_mgr_id`?
- Is the `resource_id` in the message a resource I manage?
- Is the action in the message an action performed by the specified resource?
- Is the `controller_id` in the message authorized to perform the resource action?

If all of the conditions are true, the Resource Manager sends a challenge message to the Controller's authbot.

5. The Controller's authbot digitally signs the challenge and returns a `signature_verification` message to the Resource Manager.

6. The Resource Manager verifies the signature in the `signature_verification` message.

7. With a valid signature verification the Resource Manager sends a digitally signed `resource_message` to the specified Resource (garage door opener).

8. The Resource (garage door opener) verifies the digital signature on the `resource_message` and if valid, performs the specified action (triggers the garage door opener).

9. The Resource (garage door opener) sends a status message to the private room indicating that the requested action was performed.

Exceptions:

If any of the checks mentioned in the flow fail, the flow terminates with an appropriate error message to inform participants of the failure.

Requirements:

The following requirements must be met before executing the use case:

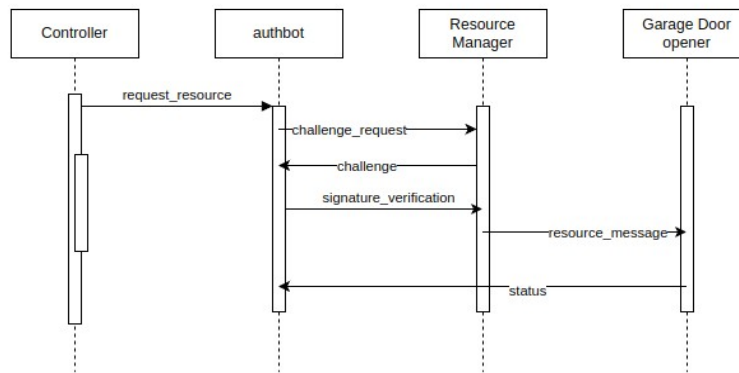
1. The Resource (garage door opener), and the specified Resource *action*, must be registered at the specified Resource Manager.
2. The Controller (*controller_id*) must be authorized to perform the specified Resource *action* at the Resource Manager.



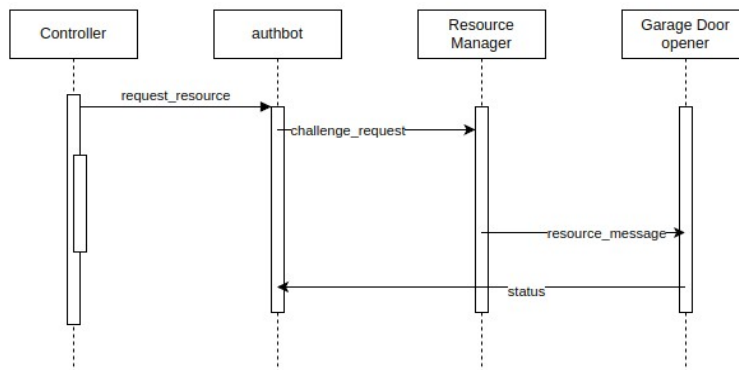
Garage Door Opener Prototype-The device with the real bright LED at the top is a Raspberry Pi, the smaller device with the LED on below it is a 5V relay module, and the device with the black and red wires at the bottom is a garage door opener. The Raspberry Pi runs the software that controls the opening and closing of the garage door via the MoneroAuth protocol.

Sequence diagram

Garage Door opener Without an active session



Garage Door opener With an active session



Garage door opener QR Code



Encoded information:

```
{"json":"2.0","method":"qr_mpcr","params":  
{"resource_mgr_id":"498EM2vdJRSV6LcRUadS7TE4BdpusMz4wWMAM8YoBAw3M8D3Zkdv  
YSQN42FBm1aG7X8pRkEFpgvZBPAh78xbYLnj1NZbgJD",  
"resource_mgr_muid":"@demo5:authbot.org","resource_id":"49FJQ4hDcNF1ebwCmxu8yG  
PnaTLGoKPYDRYS9PGrV4CkbFErBPAE8AdVB8UMMYgXLGTUedWk23o7dcxUPybLgG6QC6p  
eRkc",  
"resource_muid":"@garage-door:authbot.org","action":"trigger"}}
```