Schnorr Protocol Fob

Michael Gamota
Pedro Ocampo
Vasav Nair

Introduction

Problem

Current car fobs and garage door fobs are susceptible to different types of attacks. One common method for code generation includes the use of a counter, meaning both devices have an agreed upon “next code”, however this makes them susceptible to rolling jam attacks. This attack involves a malicious third party intercepting(jam and store) a valid unlock/open signal sent by a button fob. When the user tries to unlock the device again, the signal is again intercepted, but the third party sends the first intercepted code to the receiver. Now, the third party has the next valid code. Garage doors may use either a fixed code or have a counter based code, which makes them susceptible to replay attacks or rolling jam attacks, respectively.

Cars with passive fobs can be stolen using relay attacks. Passive fobs require the presence of a scanner, which emits EM radiation to power the passive fob which returns a code wirelessly. With a passive fob, a malicious actor just needs to artificially extend the reader(located on the exterior of a car) to trigger a response from the passive fob.

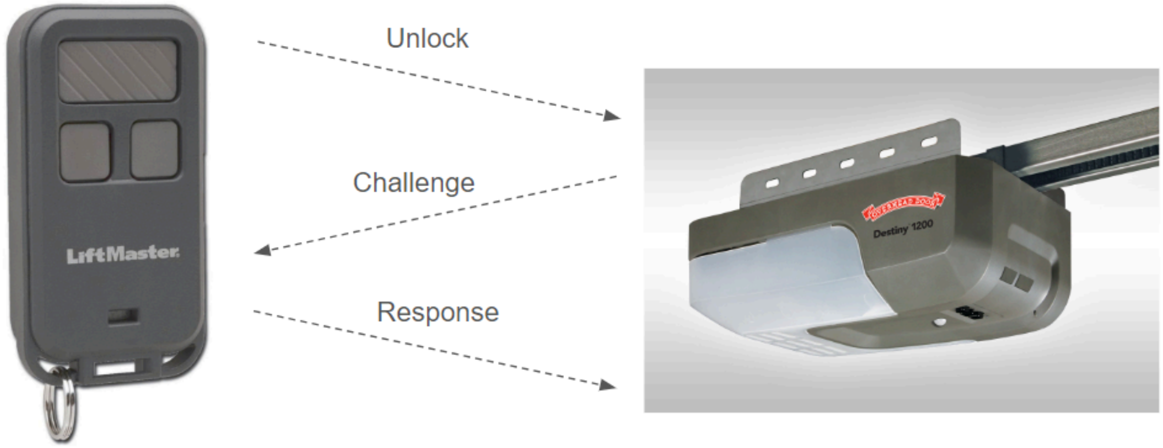
Therefore, all of these fob technologies have vulnerabilities which are relatively easily exploitable. We find this to be unreasonable as access to one’s house or car is something that should be protected with the highest level of security.

Solution

Our solution is to design the hardware for a key fob(prover) and verification and control(verifier) unit which will leverage Schnorr protocol, an authentication scheme which uses public key encryption to implement an interactive zero-knowledge proof. What this means is that, first, the fob(prover) will have a public key and private key, which can be thought of as a known serial number, and an associated secret serial number. Second, there will be an interaction between the key fob and the verification and control unit, the fob will send a message to the lock/opener, announcing its public key and the desired command. The lock/opener will respond with a “challenge”, which is a random number. The key fob will then compute the response to the “challenge” which requires knowledge of the secret key. This response is sent back to the lock/opener which can then mathematically verify if the fob has disclosed its true public key in the first message. If the public key is verified as honest and it is on a preprogrammed list, the verification and control unit will unlock the car/open the garage door. The last element of Schnorr protocol is that it is a zero-knowledge authentication scheme, which means that no information about the secret key can be gained by a third party listening in or the verification and control unit.

For the purposes of our final demonstration, the verification and control unit will be connected to a small motor which will spin when a valid exchange occurs.

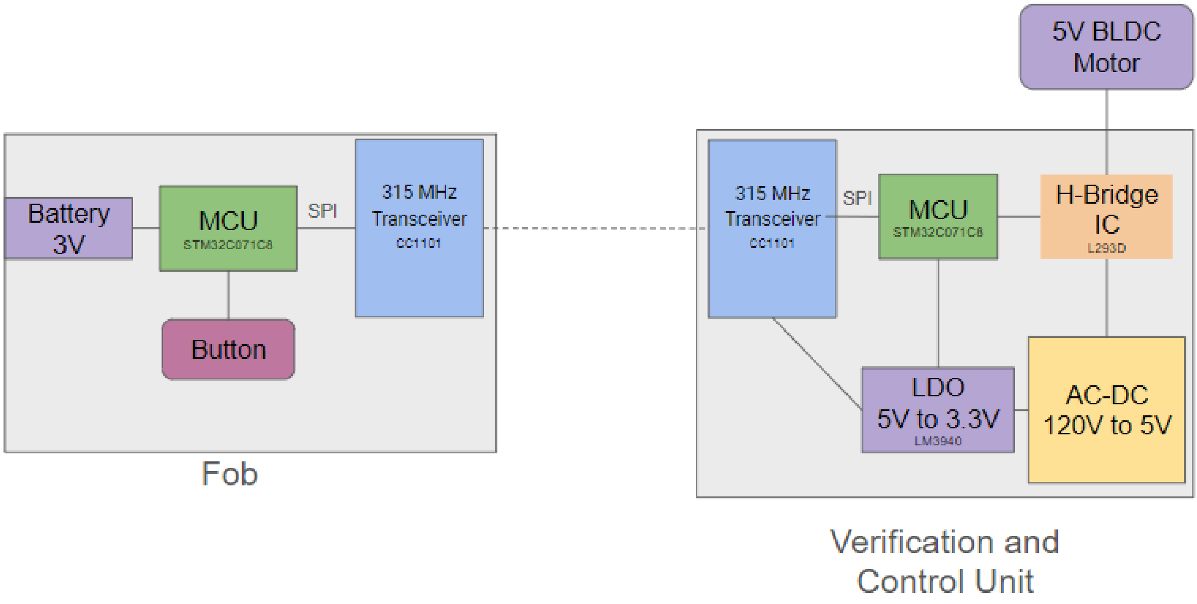
Visual Aid



High Level Requirements

- The signals must be detected and received from 3 meters and the signal integrity must be good enough that messages are able to be authenticated.
- The time between the unlock signal being sent and the motor spinning must be less than 2 seconds.
- A replay attack is not successful

Block Diagram:



Fob Subsystem

Here, our fob system will communicate with our Verification and Control Unit by sending a random number from the fob to the lock, which will verify whether it matches with its own key, and unlock in response to it in a reasonable amount of time. This fob will include a button, MCU, a 3V battery, and a 315 MHz Transceiver.

- **MCU:** We will be using an STM32C071C8 for our MCU. This will be responsible for memory allocation, creating a random 256 bit number that will be used to calculate the initial message to be sent through the RF transceiver to the verification and control unit. The MCU will be activated by a push button, and powered by a 3V coin cell battery.
    - Requirement 1: Capable of creating a 256 bit random number via different algorithms, C, and C++ functions, possibly running a hash function on ADC noise
    - Requirement 2: Capable of performing exponential calculations to calculate the valid response based on the received challenge

- **Transceiver:** We will be using a 315 MHz Transceiver CC1101 for our Transceiver subsystem. It will be responsible for receiving our 256 bit key from our MCU, which will be sent wirelessly to our Verification and Control Unit.
    - Requirement 1: Capable of receiving a 256 bit packet from MCU over SPI and receiving a packet from the Verification and Control unit over 315 MHz.
    - Requirement 2: Capable of sending a 256 bit packet to our Verification and Control unit at 315 MHz
    - Requirement 3: Antenna works at a range of 3m

- **Button:** We will be using this to activate the MCU so that our STM32C0 isn’t constantly on and draining power. Additionally once our button is pressed, the Schnorr identification protocol will initiate and the transceiver will transmit our initial packet of data.
- **Battery**: We will be using a 3 volt battery which should provide more than enough power to all of the components on our Fob subsystem.

Verification and Control System

Our verification and control unit will act as the verifier in this hardware implementation of the Schnorr Identification Protocol. This subsystem is responsible for authenticating the fob sending the message and then spinning the motor given a valid response. This subsystem will include a 315 Mhz transceiver, MCU, an AC-DC converter, and an H-bridge IC to control a 5V BLDC motor.

- **MCU:** We will be using an STM32C071C8 for our MCU. This will be responsible for memory allocation, creating a random 256 bit number that will be used as the challenge for the fob to respond to. The MCU will be powered via an LDO, which is powered by a 5V AC-DC converter.
    - Requirement 1: Capable of creating a 256 bit random number via different algorithms, C, and C++ functions, possibly running a hash function on ADC noise
    - Requirement 2: Capable of performing exponential calculations to verify the fob’s response based on the given challenge

- **Transceiver:** We will be using a 315 MHz Transceiver CC1101 for our Transceiver. This subsystem will be responsible for receiving our 256 bit initial commitment from our MCU, responding with the challenge, and receiving the fob’s response to the challenge.
    - Requirement 1: Capable of receiving a 256 bit packet from MCU over SPI and wirelessly from fob.
    - Requirement 2: Capable of sending a 256 bit packet to our fob at 315 MHz
    - Requirement 3: Antenna works at a range of 3m

- **AC-DC Converter:** Since this unit will be on the garage door motor, we should have access to a 120V AC power source. We will use an AC-DC converter to convert the Ac signal to 5V DC to power the H-bridge IC and also to a 3.3V LDO.

Tolerance Analysis: One aspect of our project that may be difficult to get right is the antenna design and RF matching needed for successful communication between the fob and the verification and control unit. To design an antenna we will use Ansys HSFF software to simulate different designs.

Ethics and Safety

Potential Ethical breaches:

We are not concerned about any possible ethical breaches during our project as our project is meant to prevent them by protecting key fob security.

Potential safety concerns:

Energy Converter: Since we are using an AC-DC converter, we need to make sure our converter properly converts the 120V AC, typical for wall outlets, to our desired voltage required for our garage door lock to function properly. The only possible safety concern is if we incorrectly build our converter, but with proper failsafes (designing fuses and grounds into our pcb) and schematics that concern will be addressed. We also need to make sure that our model garage door locks and unlocks in a reasonable amount of time, not taking too long so that if the user needs to access, it unlocks quickly.

References

[1] Just, M. (2011). Schnorr Identification Protocol. In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer [Online]. Available: https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_95
[Accessed:Sept. 19, 2024]

[2] STMicroelectronics, “STM32C071x8/xB”, Rev. A, Sept. 2024 [Online]. Available: https://www.st.com/resource/en/datasheet/stm32c071c8.pdf

[3] Texas Instruments “Low-Power Sub-1 GHz RF Transceiver”, Revision SWRS061I, Nov. 2013 [Online]. Available: https://www.ti.com/lit/ds/symlink/cc1101.pdf?ts=1726733211839&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC1101%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Depd-null-null-GPN_EN-cpc-pf-google-wwe%2526utm_content%253DCC1101%2526ds_k%253D%257B_dssearchterm%257D%2526DCM%253Dyes%2526gad_source%253D1%2526gclsrc%253Dds