HW 1: UART Security Evaluation

## Terminology (10 Pts):

1. What is Risk?

RISK is a vulnerability and a threat that are used in security to quantify outcomes or how much effort is needed. Vulnerability measures how likely an attacker is to succeed, and threat measures the outcome if the attacker did succeed or what would be lost.

1. Give an example of Confidentiality in security.

Confidentiality in security is keeping sensitive information secret, and an example of this would be when information is encrypted between users to prevent outside individuals from receiving the same information.

1. Give an example of Integrity in security.

Integrity in security is ensuring that the system cannot be modified so that the system maintains proper outputs even when under attack. An example of this would be hashing a password so that if the password or information is tampered with, it will not match the original so it is known something has been altered.

1. Give an example of Accessibility in security.

Accessibility in security is the ability of the system to be accessed by users and attackers. An example of this would be having a login page where only users can access the data if they put in their identification information.

## Security Methods (70 Pts):

1. Is a default (no implemented defenses) UART console low or high risk? Why?

A default UART console is high risk since there is no type of authentication and data is transmitted in plaintext. However, it does have the benefit of requiring physical access in order to gain control which can lower the risk factor, but it’s still really easy to gain control of the system once physical access is possible.

1. Would this be true for other methods of serial communication such as I2C or a CAN Bus? Why?

An I2C is a two-wire serial bus protocol between the MCU and sensor that requires two signal connections to the MCU where multiple devices are allowed on the two connections and get acknowledgment that signals have been correctly received. However, it is typically not automatically secure as with physical access, the data can also be manipulated or read since it’s also transmitted in plaintext similar to UART.

A CAN bus provides multi-master, message-oriented protocol that lacks built-in security but physical access is also needed to access the bus.

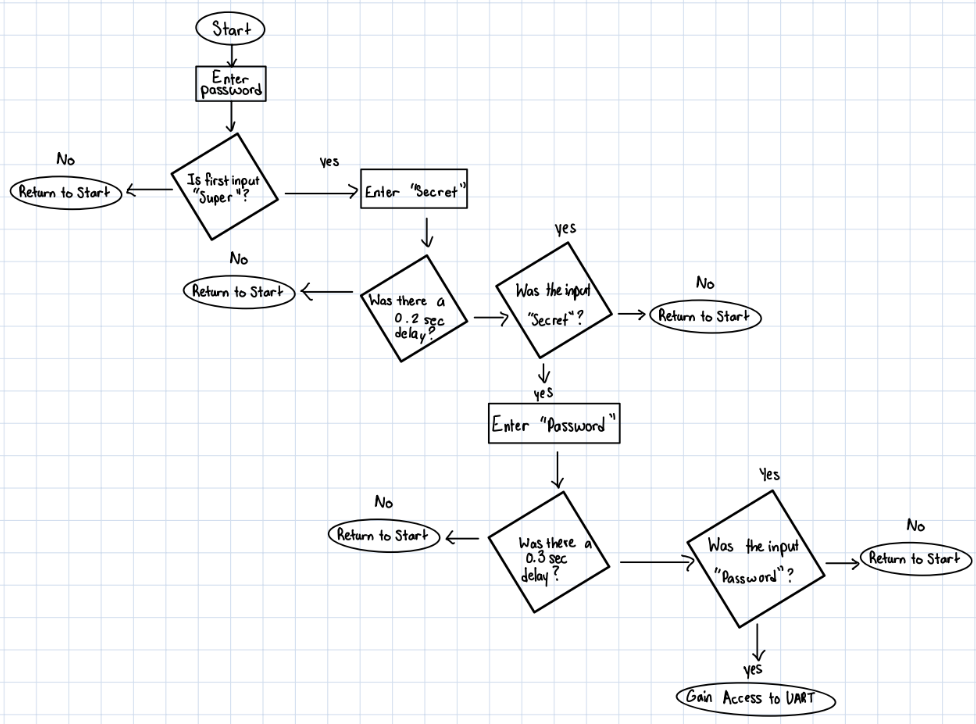
Overall, both could have similar risks to UART if there’s no implemented encryption or authentication since they also require physical access.

1. How long would it take to brute force an AES 128 bit password (assuming 1 guess per second)? How many guesses per second would be required to brute force it in 30 minutes?

Time to brute force = [(number of potential characters)^(password length)]/(guess per second) = ((2 characters)^(128 length))/(1 guess per second) = 3.402823669209385 X 10^38 seconds

Guesses per second = [(number of potential characters)^(password length)]/(Time to brute force) = [(2 characters)^(128 length)]/(30 minutes \* 60 seconds per minute) = 2.42270793885716 x10 ^19 guesses per second.

1. Please create a flowchart that describes the ‘Timed Knock’ process.



1. Create a table (or diagram) recording the state of four inputs (A, B, UART\_RX, and CLK) that describes the unlocking of a logic locking implementation. This implementation should require A and B switch logic state each clock cycle (With A high at the start and B low) and have a 2 byte password ‘F01D.’ The UART connection uses a single Stop bit with no Parity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLK Cycle | A | B | UART\_RX | Description |
| 1 | 1 | 0 | Start (No data) | State before data is received. |
| 2 | 0 | 1 | 11110000 (F0) | First byte received on UART\_RX |
| 3 | 1 | 0 | 00011101 (1D) | Second byte received on UART\_RX |
| 4 | 0 | 1 | Stop (No data) | End of logic locking implementation. |

## Closing Thoughts (20 Pts):

1. What is the best (most cost effective) way to secure a UART Console on an IoT Device.

The best and most cost-effective way to secure a UART on an IoT device would to be restrict physical access to the UART. In other words, lock the UART in a place that doesn’t allow any of its ports to be physically accessed.

1. If you were given a task to secure a low cost IoT temperature, humidity, and weather sensor that would be placed on rooftops across a city, how would you go about securing them? Note that no ‘right’ answers are here, but please describe the reasons behind each choice.

First, there would be physically securing them, since they are out in the open and are exposed to nature, it would be important to make it doesn’t get damaged. Additionally, since it is exposed, people can have physical access to the device and can tamper with it. Then I would encrypt it, so that if there is physical access or intercepted access online, it would be harder to intercept the data to read or tamper with. Next would be to add authentication to restrict who can access the data. Finally, monitoring the device to detect anything out of the ordinary to indicate a threat or action needed.