**CEG 4980 WHITE PAPER**

|  |  |
| --- | --- |
| **Name: Josiah Schmitz** | **Major: CS** |

**Death Star Plans Disseminator**

**I. Problem Statement**

The Empire has built a battle station (known as the Death Star) that is capable of wiping out entire planets with a single shot (Death Star). The rebels need to find a way to obtain plans for the Death Star in order to destroy the station. These plans, which indicate vulnerabilities for the battle station, are being held on a Raspberry Pi in an Empire server room. The plans themselves are contained in 10 PNG files along with 90 other unrelated images on the Pi.

These plans must be retrieved by a Rebel engineer with access to a Rebel server running on a laptop. He must be able to transmit the plans from the Raspberry Pi to the server without access to Wi-Fi, wired LAN, or Bluetooth. Transmission must take 600 or fewer seconds in total, so as to not make any Empire guards suspicious. Additionally, any transmissions that do occur between the Raspberry Pi and the server need to be fully encrypted.

**II. Existing Solutions**

There are a variety of existing ways to transmit the plans. For instance, the Rebel engineer can use a specific model of Raspberry Pi that has Wi-Fi capabilities, such as a Raspberry Pi Zero W, to transmit the files via Wi-Fi (Buy A Raspberry Pi Zero W – Raspberry Pi).

With a device such as this, a service called PiTunnel can be installed onto the Pi. PiTunnel will then allow the Raspberry Pi to connect to the server with SSH. Once connected, SFTP (secure file transfer protocol) can be used to directly transfer the images via Wi-Fi from the Raspberry Pi to the server.

The main reason that this solution is not suitable is that it uses Wi-Fi as its main method of transmission. There is a strict ban on using Wi-Fi to transmit the plans for the sake of this project.

**III. Proposed Solution**

The proposed solution would be to use QR-code generation to transmit the files from the Raspberry Pi to the Rebel server as encrypted data. This would be done by having the Raspberry Pi run a script that detects red circles within the 100 given images, such as through OpenCV and its Hough Circle Transform functionality (Hough Circle Transform). It would then compress the 10 identified plan images in a single ZIP archive and encrypt the folder using AES encryption (*AES Encryption & Decryption*). Finally, the Raspberry Pi would encode the encrypted data as a series of QR codes on a monitor connected to the Raspberry Pi (Divan).

The rebel server can then use either its integrated camera or a separate camera connected via USB to read the QR-codes as a stream and reconstruct the ZIP archive using the AES encryption key. To verify that the data is intact and correct, the server can display a single QR-code with the MD5 checksum of the images to be read and verified by a camera connected to the Raspberry Pi.

**CEG 4980/EE4910 WHITE PAPER**

**IV. References**

*Buy A Raspberry Pi Zero W – Raspberry Pi*, www.raspberrypi.com/products/raspberry-pi-zero-w/. Accessed 5 Sept. 2025.

*AES Encryption & Decryption in Python: Implementation, Modes & Key Management*, onboardbase.com/blog/aes-encryption-decryption/. Accessed 5 Sept. 2025.

“Death Star.” *StarWars.Com*, www.starwars.com/databank/death-star. Accessed 5 Sept. 2025.

Divan. “Divan/Txqr: Transfer Data via Animated QR Codes.” *GitHub*, github.com/divan/txqr. Accessed 5 Sept. 2025.

“Hough Circle Transform.” *OpenCV*, docs.opencv.org/3.4/d4/d70/tutorial\_hough\_circle.html. Accessed 5 Sept. 2025.