# ENGR 13300

# Python Group Project Fall 2021

**Project Description:**

With a large scale of image data rapidly generated and transferred between different devices and an increased number of cyberattacks, the security of the data is of great concern. This has led to the study of image encryption, digital signature, and watermarking methods. Image encryption is widely used in social media, telemedicine, space communications, etc., where sensitive data need is involved.

At the same time, as the amount of image data explodes, the demand for reliable techniques for computers to extract meaningful and valuable information from the images increases. Nearly all the engineering disciplines nowadays use image analysis techniques to implement solutions to engineering problems. For example, computer vision is used to find small features as microstructural damage in materials to large objects like Supernova or blackholes. Image analysis and computer vision are widely used for identifying astronomical objects that are already known based on their shape, spectrum, reflectivity, and other features to guide a deep-space probe. Computer vision also played a major row in position estimation for landers and navigation for rovers on Mars and Moon.

**In this project, you will develop an XOR cipher in Python to encrypt and decrypt images and develop an algorithm to detect the features of interest in the images based on the gradient map of an image.** The goal of this project will help you better understand the general libraries, image encryption process, and basic feature detection methods.

**Requirements:**

1. You will be only allowed to use `numpy`, `matplotlib`, and `scipy`.
2. **Utilize function(s) in this project to allow for modularity and reusability.** This will also allow your project team to assign each group member with a specific task, and the member will individually work on their task and build their user-defined function that accepts inputs they need and return the desired output.
3. You will be mainly working with the image called 'Pale_Blue_Dot_Encrypted.tiff'. You can also find sample images on Brightspace. Those will be helpful for debugging your program.

**Deadlines:**

In class **demo: October 19th**

Final Reports and Programs **uploaded to Gradescope, Tuesday, October 19th at the start of class**.

Update Notes:

You will need to save your images in a tiff format in this project. Comparing to PNG, JPG formats, TIFF files don't compress the image and store the exact pixel values. The image encryption process requires the exact pixel values of the original images. Slightly change of the pixel value will result in a failure in image decryption.

## Part 1: Image importation

In the first part of this project, you will develop a Python program to import the image. You can find ways to import images using `matplotlib` in the resources provided. Your program should:

- **Prompt the user for the name of a color image file**.
- **Check the data type and the range of the image data. The data type of the image should be uint8 integers, which range from 0 to 255. You will need to convert the image data to the correct range and make sure the image data is in the correct data type.**
- **Pass the image data to other UDFs to perform future tasks.**

Try to answer the following questions in your report:

- **What are the data type and the range of the data types if you try to import a PNG file using `matplotlib`? How about the JPG image? How about the TIFF image?**
- **If the data type of the image data if was not of the unit8 type, how did you convert that?**

## Part 2: Image decryption

In this part, you will develop a key generator and an XOR cipher to decrypt the images given. We will be using what is called a symmetric-key cryptosystem, as illustrated in Figure 1.

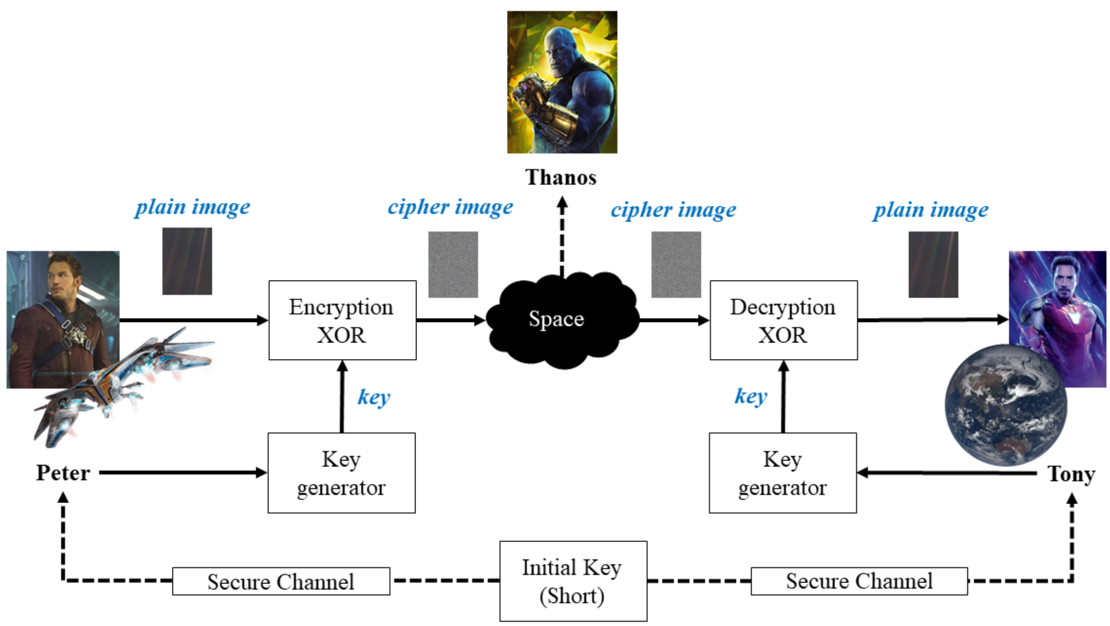
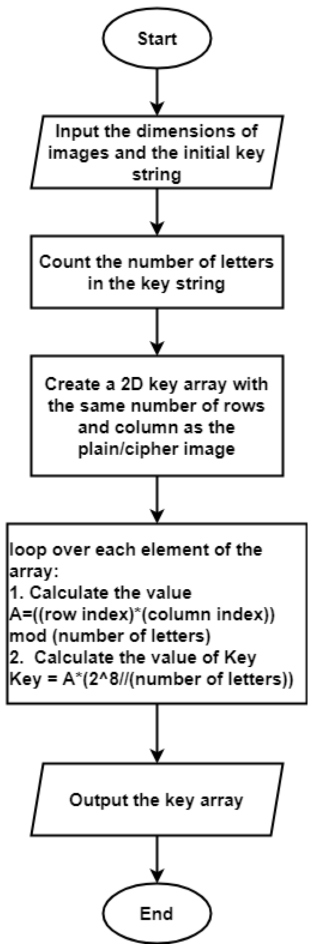



**Figure 1. Symmetric-key cryptosystem**

In this symmetric-key cryptosystem, there are two users, Peter and Tony, who will communicate over an insecure channel (the deep space communication channel in this case). The channel is considered insecure because Thanos, the bad guy, also has access to the channel, or he can hack into the system to receive the radio wave. To solve this problem, Peter encrypts the image using a symmetric algorithm and generates a cipher image sent to Tony.  And Tony is able to decrypt the image using the same algorithm and key to reconstruct the plain image. The key to the image is generated from the initial key, which usually consists of a few letters or symbols and is shared through a secure channel. Tony and Peter will use the same key generator to generate keys for a particular image.

In this part, you are first going to program a simplified key generator that takes the dimensions of the plain/cipher image and a short message string (the initial key) as the inputs. The key you generated should be a two-dimensional matrix with the same numbers of rows and columns as the input images.

The key generator should follow the flow chart below (row and column index starting from 0):




The initial key is

```
COME AND GET YOUR LOVE
```

It's the name of the song by rock band Redbone, released in 1973 and was featured in the movie *Guardians of the Galaxy* in 2014.

After you finish the key generator, you will need to program an XOR cipher. The XOR cipher will take the encrypted image and the key array as the input and generate the plain image. You can find more information about the XOR cipher in the resources file.

Check your program with the example images provided before you move forward.

In this part, your program should:

- **Prompt the user for the initial key**.
- **Correctly generate the key array for different sizes of images with the initial key string given. The data type of the key array should be uint8 integers, which range from 0 to 255.**
- **Use the XOR cipher to decrypt all three color channels of the cipher image to get the plain image.**
- **Output and save the plain image in the .tiff format as the cipher image.**

Try to answer the following questions in your report:

- **Did your XOR cipher correctly decrypt the example images given to you? How did you check that?**
- **How did you design your XOR cipher? What loop did you use? What functions from the libraries did you use?**
- **What's the running time of your XOR cipher to decrypt the images? Try to show the relationship between running time and the number of pixels.**
- **What are the pros and cons of the XOR cipher?**

### Part 3: Feature detection

After you finish Part 1 and 2, you should be able to get an image similar to Figure 2. You may find a tiny blue dot in one of the rays, which is the earth. Your goal in this part of the project is to find the exact location of the blue dot using the gradient map of the image. To create the gradient map, you will first convert the color image to a grayscale image and smooth out the noise using a Gaussian filter. Then calculate the gradients on x and y directions using the Sobel operator. You will need to output the magnitude of the gradient as a grayscale image. Make sure you convert the image data type to `numpy.float64` when you are performing the operations above.

**Figure 2. This updated version of the iconic "Pale Blue Dot" image taken by the Voyager 1 spacecraft Credits: NASA/JPL-Caltech**

Based on the gradient map, you will need to develop your own method and algorithm to find the exact location of the earth.

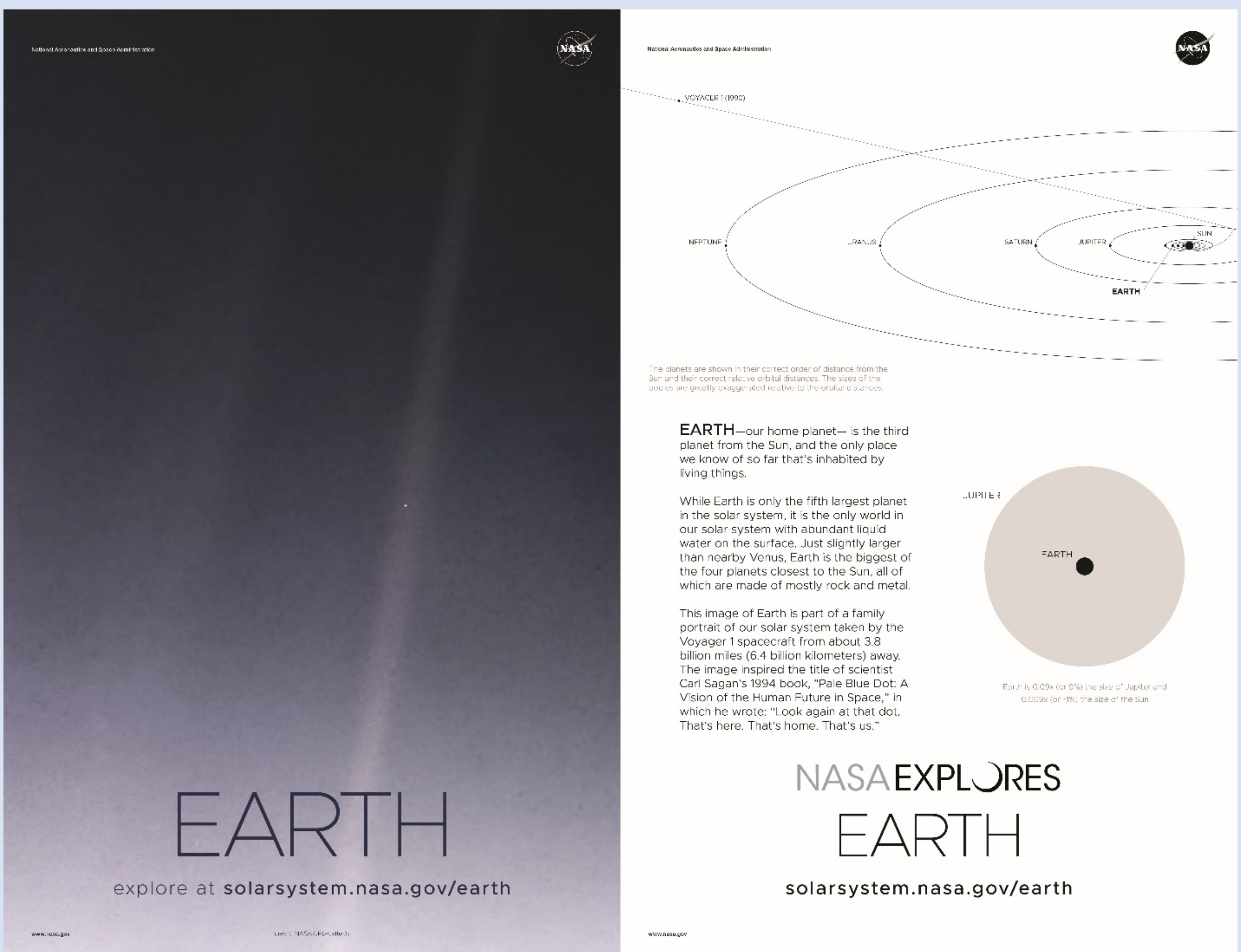
In this part, your program should:

- **Convert the image data from** `numpy.uint8` **to** `numpy.float64`
- **Output the gradient map of the image as a greyscale image.**
- **Output the row and column index (either float or integer, depends on your algorithm) of the pixel where the earth locates in the image.**
- **Output a cropped image with a size of 101 pixel * 101 pixel with the earth image at the center.**

Try to answer the following questions in your report:

- **What's the range of the gradient map?**
- **How did you find the exact location of the earth? Is there any limitation of your algorithm?**
- **Did the Gaussian filter (smoothing process) affect your result?**

## Background story of the image: Pale Blue Dot



The famous photo of the Pale Blue Dot was taken by Voyager 1 on the Valentine's Day in 1990. As it was about 4 billion miles away from the earth, it turned around for the one last time to take a glance of our home planet. The earth, a pale blue dot with a size smaller than 0.12 pixel, was lying quietly in the ray of the Sun.

Carl Sagan, an American astronomer, who suggested Voyager 1 to take this image as it departs for the fringes of the solar system, wrote the following inspiring words in his book *Pale Blue Dot*:

> *Look again at that dot. That's here. That's home. That's us. On it everyone you love, everyone you know, everyone you ever heard of, every human being who ever was, lived out their lives. The aggregate of our joy and suffering, thousands of confident religions, ideologies, and economic doctrines, every hunter and forager, every hero and coward, every creator and destroyer of civilization, every king and peasant, every young couple in love, every mother and father, hopeful child, inventor and explorer, every teacher of morals, every corrupt politician, every "superstar," every "supreme leader," every saint and sinner in the history of our species lived there--on a mote of dust suspended in a sunbeam.*
>
> *The Earth is a very small stage in a vast cosmic arena. Think of the rivers of blood spilled by all those generals and emperors so that, in glory and triumph, they could become the momentary masters of a fraction of a dot. Think of the endless cruelties visited by the inhabitants of one corner of this pixel on the scarcely distinguishable inhabitants of some other corner, how frequent their misunderstandings, how eager they are to kill one another, how fervent their hatreds.*
>
> *Our posturings, our imagined self-importance, the delusion that we have some privileged position in the universe, are challenged by this point of pale light. Our planet is a lonely speck in the great enveloping cosmic dark. In our obscurity, in all this vastness, there is no hint that help will come from elsewhere to save us from ourselves.*
>
> *The Earth is the only world known so far to harbor life. There is nowhere else, at least in the near future, to which our species could migrate. Visit, yes. Settle, not yet. Like it or not, for the moment the Earth is where we make our stand.*
>
> *It has been said that astronomy is a humbling and character-building experience. There is perhaps no better demonstration of the folly of human conceits than this distant image of our tiny world. To me, it underscores our responsibility to deal more kindly with one another, and to preserve and cherish the pale blue dot, the only home we've ever known.*

From the dawn of human history, humans have always looked up to the sky and dreamed about space. From Carthaginians to Han Chinese, from the Paranal Observatory in Chile to the RATAN-

600 facing the clear northern sky in Zelenchukskaya, from the FAST (Five-hundred-meter Aperture Spherical Telescope) in Guizhou China to the Hubble Telescope in space, from Yuri Gagarin to Valentina Tereshkova, from the very first survivable landing on the moon by Luna 9 to the first human footprint by Neil Armstrong, and, of course, from Gustav Holst's orchestral suite, The Planets to the Little Prince from a tiny planet in Saint-Exupéry's fairy tale, people from all walks of lives of any civilization at any time at any place tried to explore and to understand the universe with their best techniques and effort.

We always draw new insight and better understanding of where we are and how the world looks like every mile away from this adorable blue planet. We came a long way and have achieved great things and will continue to do that. One day, you, who are doing this course project right now, will become part of that history. You may become the Chief Engineer of NASA or the first man or woman on Mars. But no matter how far you go, this photo, the Pale Blue Dot, is a constant reminder of who we are in the history of the universe, just like Immanuel Kant said in his book *The Critique of Practical Reason*:

> *Two things fill the mind with ever new and increasing admiration and awe, the oftener and the more steadily we reflect on them: the starry heavens above and the moral law within.*

## Part 4: Image encryption

In this part, you will generate a new key generator to make it more difficult for Thanos to crack the cipher image. Thanos will use the pixel intensity distributions of the three channels (and the help of Infinity Stones) to get the original image. Try to plot the histograms of the intensity distribution of three color channels using Python and try to answer the following questions before designing a key generator:

- **Did you see peaks and other characteristics in the histograms?**
- **What type of distribution of the pixel intensity do you expect to see if the key and key generator are considered as perfect? Why?**
- **What statistical quantity do you think can be used to tell how well the key generator is?**

Now design your key generator to increase the difficulty of cracking the image. You may find resources in the material provided to you. You can also find other resources on your own. (You can search keywords like pseudorandom numbers, chaotic map, chaos-based image encryption, etc., on Google Scholar to get more information.) Try to compare intensity histograms of the encrypted image using the original and the new key generator.

In this part, your program should:

- **Prompt the user for the new initial key**.
- **Your new key generator should output a key array with the size same as your image, and the data type should be uint8 integers, which range from 0 to 255.**
- **Encrypt the image you just decrypted using the new key and the XOR cipher.**

- **Output the encrypted image.**
- **Output the distribution of the cipher image pixel intensities using the original and the new key generator.**

Try to answer the following questions in your report:

- **How did you design your new key generator?**
- **Do you think the results are better? Why?**

## Part 5: Symmetric-key cryptosystem

After you finish Part 1 – 4, you are ready to experience the symmetric-key cryptosystem.

1. **Encryption**
   Take two images (team portrait, beautiful scenery, funny moments, or anything you are comfortable sharing. The images must be professional and appropriate. ). Try to encrypt the images using your new key generator and the XOR cipher.
   Name your encrypted image in the following way:
   Section Number_Your Team Number_Image_1.tiff
   Section Number_Your Team Number_Image_2. tiff

2. **Communicate via an insecure channel**
   Upload these two cipher images to the box folder where everyone has access.

3. **Key distribution (Communicate via a secure channel)**
   Share your initial key and key generator function with your paired team via a secure channel of your choice, where only you and your paired team have access.

4. **Decryption**
   Download the images from your paired team from the box fold. Try to decrypt the image using your XOR cipher and the initial key and key generator function you get from the secure channel. Show your image to your paired team to see whether you successfully get the plain image.

## Deliverables

Each team will submit a project report outlining their research into image encryption and decryption, the application of image processing, and the program the team developed. The project report will follow the common structure of a technical report. The Python files, along with a report, will be uploaded to Gradescope as the final submission. Make sure that the files will run as they are uploaded (e.g. have them all in the same folder). You may zip all your files together. The report will consist of:

1. **Project Motivation**

   You will need to describe how image encryption and image processing methods are used in the engineering discipline citing at least three credible sources using either APA or MLA format. This section will outline the major problems trying to be solved in these areas and the state-of-the-art methods that have been developed or widely used.

2. **Project Overview and Methods**

   In this section, you will describe the background of the image encryption/decryption, how the XOR cipher works and how the gradient map was derived and was used for feature detections. You will need to summarize the theoretical background of the technique used in this project.

3. **Discussion of Algorithm Design**

   This section will include flow charts of the overall program and each of the user-defined functions. This section should describe the design rationale for modularizing the code to meet the needs of future uses of image encryption and processing. You will need to summarize the algorithm you designed and used for Python programming. Answer the questions above in this part of your report.

4. **References:**

   List of resources used to research the application of image encryption and processing in different engineering disciplines and any additional resources used for methods for conducting image processing.

5. **Appendices: Include as appendices to your main report**
   a. **User manual** with sample inputs and outputs (including pictures).
   b. **Project management Plan**: Summary of the contributions of each member of the team to the project. Describe your team's methods for collaboration that facilitated active participation of and learning by all team members. Include a discussion of opportunities for improvement for future team projects.
   c. **Discussion of Design Process**: Description of how your team followed the design process, which is defined as "a process of devising a system, component, or process, to meet desired needs within constraints. It is an iterative, creative, decision-making process that involves developing requirements, performing analysis and synthesis, generating multiple solutions, evaluating solutions against requirements, considering risks, and making trade-offs."

d. **Code:** The team should upload the Python files, but they should also include the code as an appendix in the report. Code should be well commented to increase readability for future users of the algorithm.

**Project Grading Summary (a detailed grading rubric will be provided):**