# CS 1010

# Image Encryption and Decryption

## Another application of Raspberry Pi and PiCamera


Prof. Kartik Bulusu, CS Dept.


from picamera import PiCamera


**Teaching Assistants**:
Marshall Thompson, CS Dept.
Jonathan Garcia, MAE Dept.
Matthew Dionne, CS and EMSE Dept.

**Learning Assistants:**
Josie Libbon, CS Dept.
Josh Rizika, CS Dept.
Miles Grant, CS Dept.
Addy Irankunda, Physics Dept.
Talia Novack, CS Dept.
Fred Kamgang, CS Dept.



GW School of Engineering & Applied Science
THE GEORGE WASHINGTON UNIVERSITY

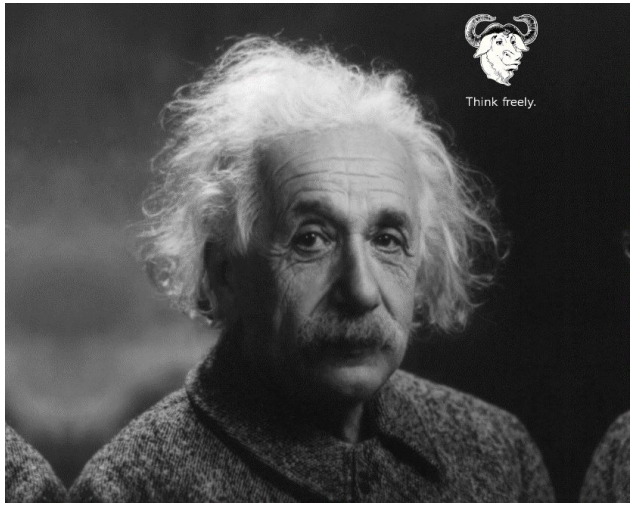
Fall 2022


Photo: Kartik Bulusu

# Very basics of data encryption-decryption and applications

**Encryption** is the **transformation of data** into some **unreadable form**.

**Decryption** is the reverse of encryption; it is the **transformation of encrypted data** back into some **intelligible form**.




$$\begin{bmatrix} 49 & 49 & \cdots & 34 & 35 & 35 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 40 & 34 & \cdots & 51 & 49 & 46 \end{bmatrix}$$

**Image encryption** - process of encoding image with the help of an encryption algorithm in such a way that unauthorized users can't access it.

**Authorization entails a "key".**

Created by Round Icons from the Noun Project

# Building up the vocabulary: Shift cipher

## Assign each letter in the alphabet a number

- Start from 0
- I have row matrix: $A_{1x26}$

## Message (X) = T E A C H

- Convert the letter into the number that matches its order in the alphabet starting from 0
- I now have a row matrix: $X_{1x5} = [19\ 4\ 0\ 2\ 7]$

## To encrypt assign a shift key (K) = 4

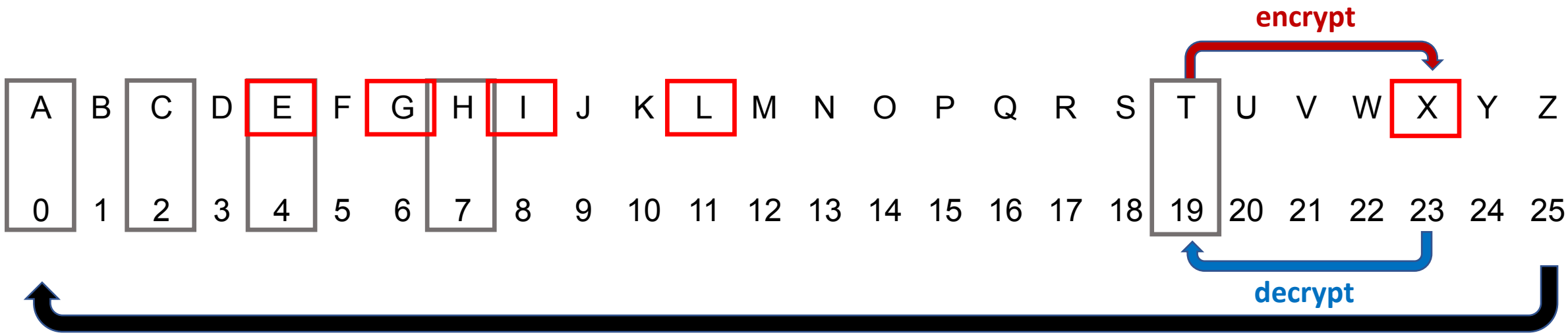
- Must be an integer from 0 to 25
- Map each letter to a different letter using the shift key
- **Y = (X+KJ) where *J is a vector-of-ones i.e., [1 1 1 1 ….]***
- I have a new row matrix: $Y_{1x5} = [23\ 8\ 4\ 6\ 11]$

## Encrypted message (Y) = X I E G L

## To decrypt apply the same shift key (K) = 4

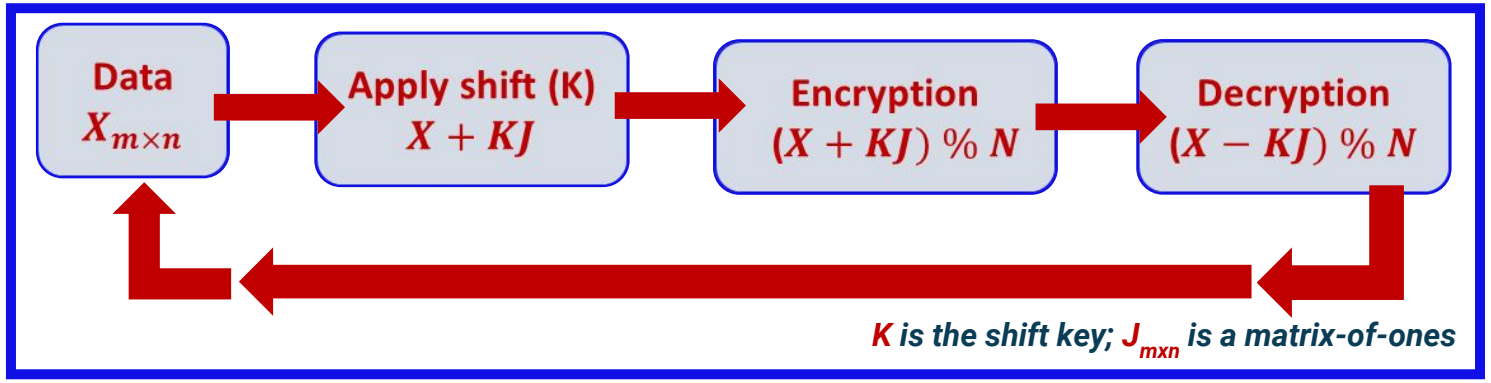
- Map each encrypted letter to a different letter using the shift key
- **X = (Y-KJ) where *J is a vector-of-ones i.e., [1 1 1 1 ….]***
- I have a new row matrix: $X_{1x5} = [19\ 4\ 0\ 2\ 7]$

## Decrypted Message (X) = T E A C H



# Building up the vocabulary: Modulo

In computing, the **modulo operation** finds the **remainder after division** of one number by another.




## Encryption

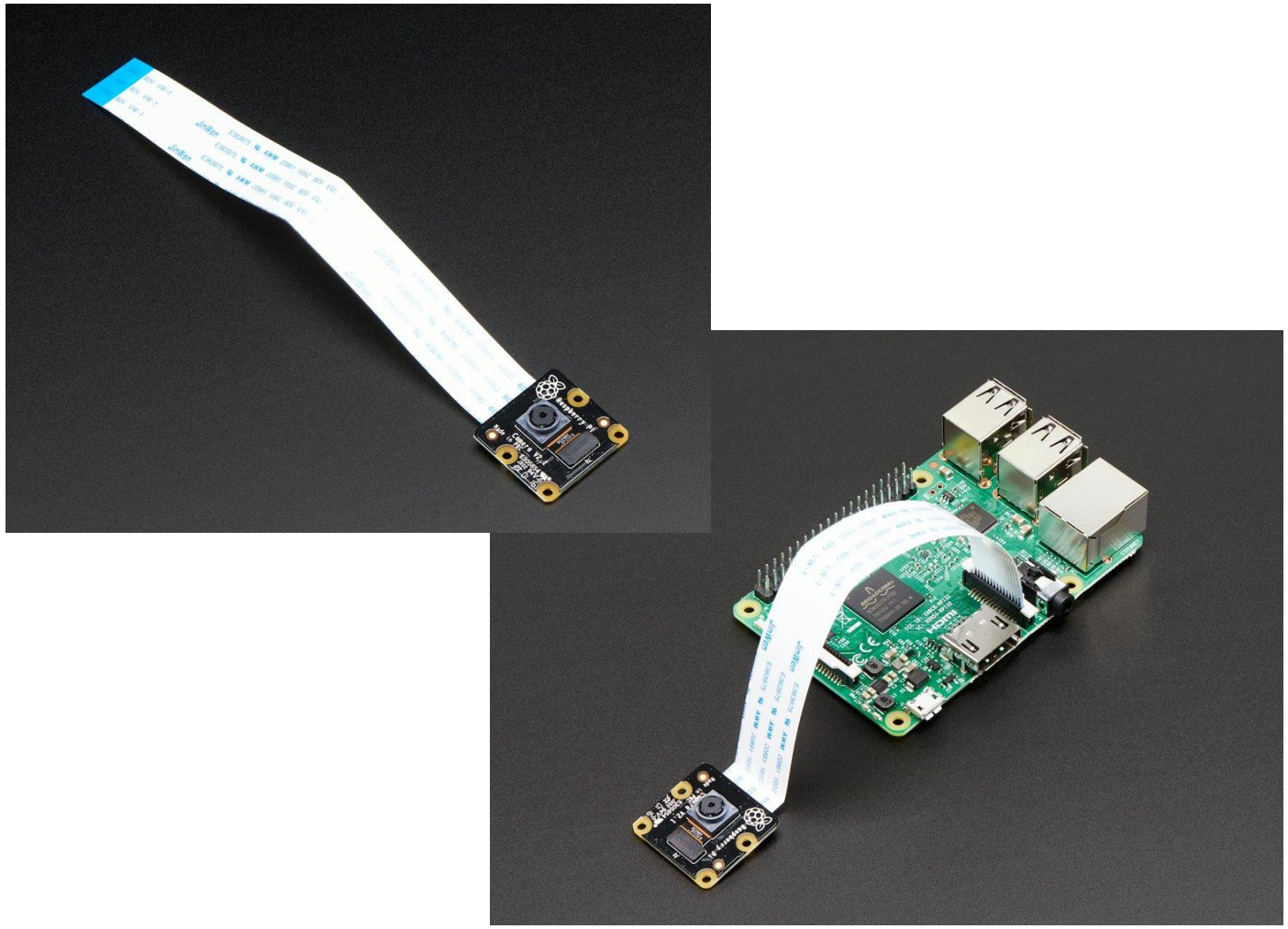
| | T | E | A | C | H |
|---|---|---|---|---|---|
| X | 19 | 4 | 0 | 2 | 7 |
| K | 20 | 20 | 20 | 20 | 20 |
| X + K | 39 | 24 | 20 | 22 | 27 |
| (X + K)%N | 13 | 24 | 20 | 22 | 1 |
| | N | Y | U | W | B |

## Decryption

| | N | Y | U | W | B |
|---|---|---|---|---|---|
| Y | 13 | 24 | 20 | 22 | 1 |
| K | 20 | 20 | 20 | 20 | 20 |
| Y - K | -7 | 4 | 0 | 2 | -19 |
| (Y - K)%N | 19 | 4 | 0 | 2 | 7 |
| | T | E | A | C | H |

**The in-class programming exercise will demonstrate these operations on images using Python**

# Introducing the Pi NoIR Camera



- 8 megapixel native resolution high quality Sony IMX219 image sensor
- 3280 x 2464 pixel static images
- Capture video at
  - 1920 x 1080 p30
  - 1280 x 720 p60
  - 640 x 480 p90 resolutions
- No Infrared (NoIR) filter
  - Infrared photographs or photographing objects in low light (twilight) conditions

**Source:**
https://www.adafruit.com/product/3100#description

## Apply Python to process the image



## Demo of Deepnote

## Import Libraries

## Import images

## Perform encryption and decryption

- Deepnote is a free online data science notebook.
- Jupyter-compatible with **real-time collaboration** and running in cloud.
- Deepnote is **built for the browser** so you can use it across any platform (Windows, Mac, Linux or Chromebook).

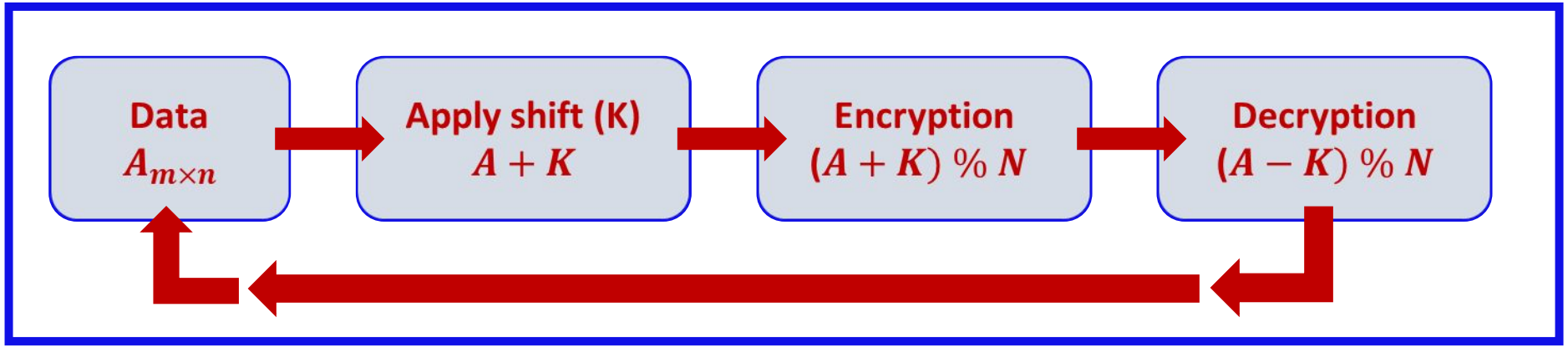
Deepnote.com

# Modulo operation – Application in data encryption-decryption

**Encryption** is the **transformation of data** into some **unreadable form**.

**Decryption** is the reverse of encryption; it is the **transformation of encrypted data** back into some **intelligible form**.

In computing, the **modulo operation** finds the **remainder after division** of one number by another.




**The in-class Raspberry Pi exercise will demonstrate these operations on images using Python**