PROJECT REPORT

on

Data Encryptor And Decryptor

(CSE III Semester Mini project)

2023-2024

Submitted to:

Mr. Ajay Shukla

Submitted by:

Mr. Abhishek Verma

Roll. No: 2218177

CSE-C2-III-Sem

Session: 2022-2026

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

Certified that Mr. Abhishek Verma (Roll No.- 2218**177**) has developed mini project on **Data Encryptor and Decryptor** for the CS III Semester Mini Project Lab in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date: 16/01/24

Mr. Ajay Shukla

Class Co-ordinator

CSE-C2-3-Sem

(CSE Department)

GEHU Dehradun

ACKNOWLEDGMENT

We would like to express our gratitude to The Almighty Shiva Baba, the most Beneficent and the most Merciful, for completion of project.

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-ordinator Mr. Ajay Shukla for his patience, support and encouragement throughout the completion of this project and having faith in us.

At last but not the least We greatly indebted to all other persons who directly or indirectly helped us during this work.

Mr. Abhishek Verma

Roll No.- 2218177

CSE-C2-3-Sem

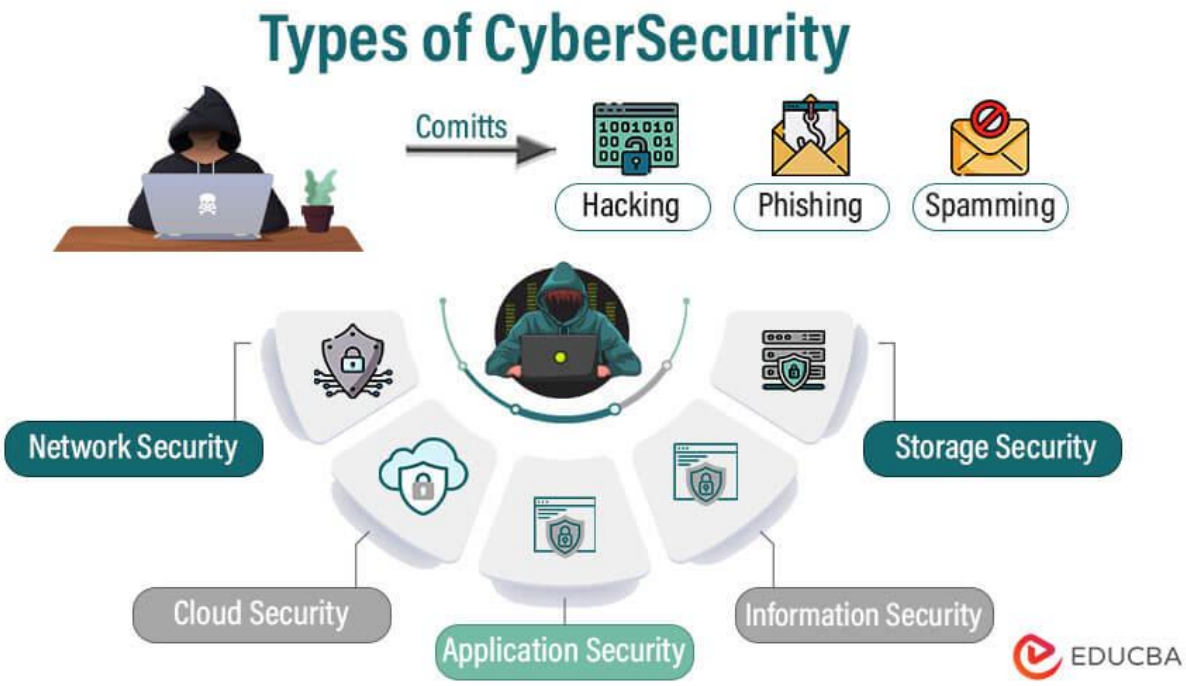
Session: 2023-2024

GEHU, Dehradun

TABLE OF CONTENTS

1. INTRODUCTION

Cybersecurity is the protection to defend internet-connected devices and services from malicious attacks by hackers, spammers, and cybercriminals. The practice is used by companies to protect against phishing schemes, ransomware attacks, identity theft, data breaches, and financial losses.



Types of cyber sercuity:

1. Application security

Application security prevents unauthorized access and use of applications and connected data. Because most vulnerabilities are introduced during the development and publishing stages, application security includes many types of cybersecurity solutions to help identify flaws during the design and development phases that could be exploited and alert teams so they can be fixed.

Despite best efforts, flaws do slip through the cracks. Application security also helps protect against these vulnerabilities.

A subset of application security is web application security. It focuses on protecting web applications, which are frequently targeted by cyber attacks.

2. Cloud security

Cloud security focuses on protecting cloud-based assets and services, including applications, data, and infrastructure. Most cloud security is managed as a shared responsibility between organizations and cloud service providers.

3. Critical infrastructure security

Special security processes and types of cybersecurity solutions are used to protect the networks, applications, systems, and digital

assets depended on by critical infrastructure organizations (e.g., communications, dams, energy, public sector, and transportation). Critical infrastructure has been more vulnerable to cyber attacks that target legacy systems, such as SCADA (supervisory control and data acquisition) systems. While critical infrastructure organizations use many of the same types of cybersecurity as other subcategories, it is often deployed in different ways.

4. Data security

A subset of information security, data security combines many types of cybersecurity solutions to protect the confidentiality, integrity, and availability of digital assets at rest (i.e., while being stored) and in motion (i.e., while being transmitted).

5. Endpoint security

Desktops, laptops, mobile devices, servers, and other endpoints are the most common entry point for cyber attacks. Endpoint security protects these devices and the data they house. It also encompasses other types of cybersecurity that are used to protect networks from cyberattacks that use endpoints as the point of entry.

6. IoT (Internet of Things) security

IoT security seeks to minimize the vulnerabilities that these proliferating devices bring to organizations. It uses different types of cybersecurity to detect and classify them, segment them to limit

network exposure, and seek to mitigate threats related to unpatched firmware and other related flaws.

7. Mobile security

Mobile security encompasses types of cybersecurity used to protect mobile devices (e.g., phones, tablets, and laptops) from unauthorized access and becoming an attack vector used to get into and move networks.

8. Network security

Network security includes software and hardware solutions that protect against incidents that result in unauthorized access or service disruption. This includes monitoring and responding to risks that impact network software (e.g., operating systems and protocols) and hardware (e.g., servers, clients, hubs, switches, bridges, peers, and connecting devices).

The majority of cyber attacks start over a network. Network cybersecurity is designed to monitor, detect, and respond to network-focused threats.

9. Operational security

Operational security covers many types of cybersecurity processes and technology used to protect sensitive systems and data by

establishing protocols for access and monitoring to detect unusual behavior that could be a sign of malicious activity.

10. Zero trust

The zero trust security model replaces the traditional perimeter-focused approach of building walls around an organization's critical assets and systems. There are several defining characteristics of the zero trust approach, which leverages many types of cybersecurity.

At its core, zero trust is based on several practices, including:

- Continuously verifying users' identity
- Establishing and enforcing the principle of least privilege for access, granting only the access that is explicitly required for a user to perform a job and only for as long as that access is required
- Microsegmenting networks
- Trusting no users (i.e., internal or external)

Many of the solutions within each of these types of cybersecurity are used across subcategories, such as:

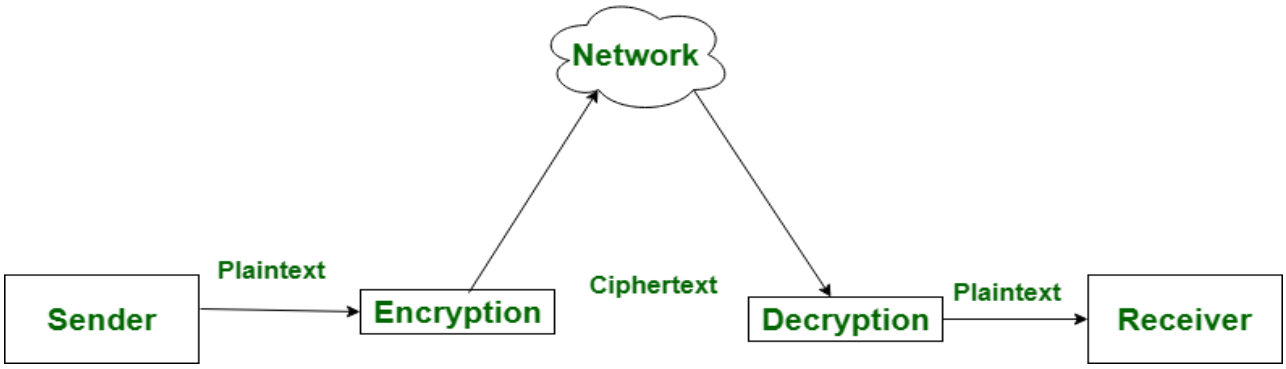
- Anti-malware software
- Antivirus systems
- Backup
- Data loss prevention (DLP)

- Enterprise mobility management
- Encryption
- Endpoint detection and response (EDR)
- Enterprise mobility management (EMM)
- Firewalls
- Identity and access management (IAM)
- Intrusion detection and prevention system (IDPS)
- Mobile application management (MAM)
- Multi-factor authentication
- Network access control (NAC)
- Next-generation firewall (NGFW)
- Secure access service edge (SASE)
- Secure email gateways (SEG)
- Security information and event management (SIEM)
- Security orchestration, automation, and response (SOAR)
- User and entity behavior analytics (UEBA)
- Virtual private networks (VPNs)
- Web application firewalls (WAFs)

Encryption and Decryption:

Encryption is the process by which a readable message is converted to an unreadable form to prevent unauthorized parties from reading it. **Decryption** is the process of converting an encrypted message back to its original (readable) format. The original message is called the **plaintext message**. The encrypted message is called the **ciphertext message**.

Digital encryption algorithms work by manipulating the digital content of a plaintext message mathematically, using an encryption algorithm and a digital key to produce a ciphertext version of the message. The sender and recipient can communicate securely if the sender and recipient are the only ones who know the key.



Types of Encryption:

There are two types of encryptions in widespread use today: **symmetric** and **asymmetric** encryption. The name derives from whether or not the same key is used for encryption and decryption.

Symmetric Encryption

Symmetric encryption is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic data. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. This encryption method differs from asymmetric encryption where a pair of keys - one public and one private - is used to encrypt and decrypt messages.

By using symmetric encryption algorithms, data is "scrambled" so that it can't be understood by anyone who does not possess the secret key to decrypt it. Once the intended recipient who possesses the key has the message, the algorithm reverses its action so that the message is returned to its original readable form. The secret key that the sender and recipient both use could be a specific password/code or it can be random string of letters or numbers that have been generated by a secure random number generator (RNG). For banking-grade encryption, the symmetric keys must be created using an RNG that is certified according to industry standards, such as FIPS 140-2.

Asymmetric Encryption

Asymmetric cryptography, also known as public-key cryptography, is a process that uses a pair of related keys -- one public key and one private key -- to encrypt and decrypt a message and protect it from unauthorized access or use.

A public key is a cryptographic key that can be used by any person to encrypt a message so that it can only be decrypted by the intended recipient with their private key. A private key -- also known as a secret key -- is shared only with key's initiator.

When someone wants to send an encrypted message, they can pull the intended recipient's public key from a public directory and use it to encrypt the message before sending it. The recipient of the message can then decrypt the message using their related private key.

If the sender encrypts the message using their private key, the message can be decrypted only using that sender's public key, thus authenticating the sender. These encryption and decryption processes happen automatically; users do not need to physically lock and unlock the message.

Many protocols rely on asymmetric cryptography, including the transport layer security (TLS) and secure sockets layer (SSL) protocols, which make HTTPS possible.

The encryption process is also used in software programs that need to establish a secure connection over an insecure network, such as browsers over the internet, or that need to validate a digital signature.

Increased data security is the primary benefit of asymmetric cryptography. It is the most secure encryption process because users are never required to reveal or share their private keys, thus decreasing the chances of a cybercriminal discovering a user's private key during transmission.

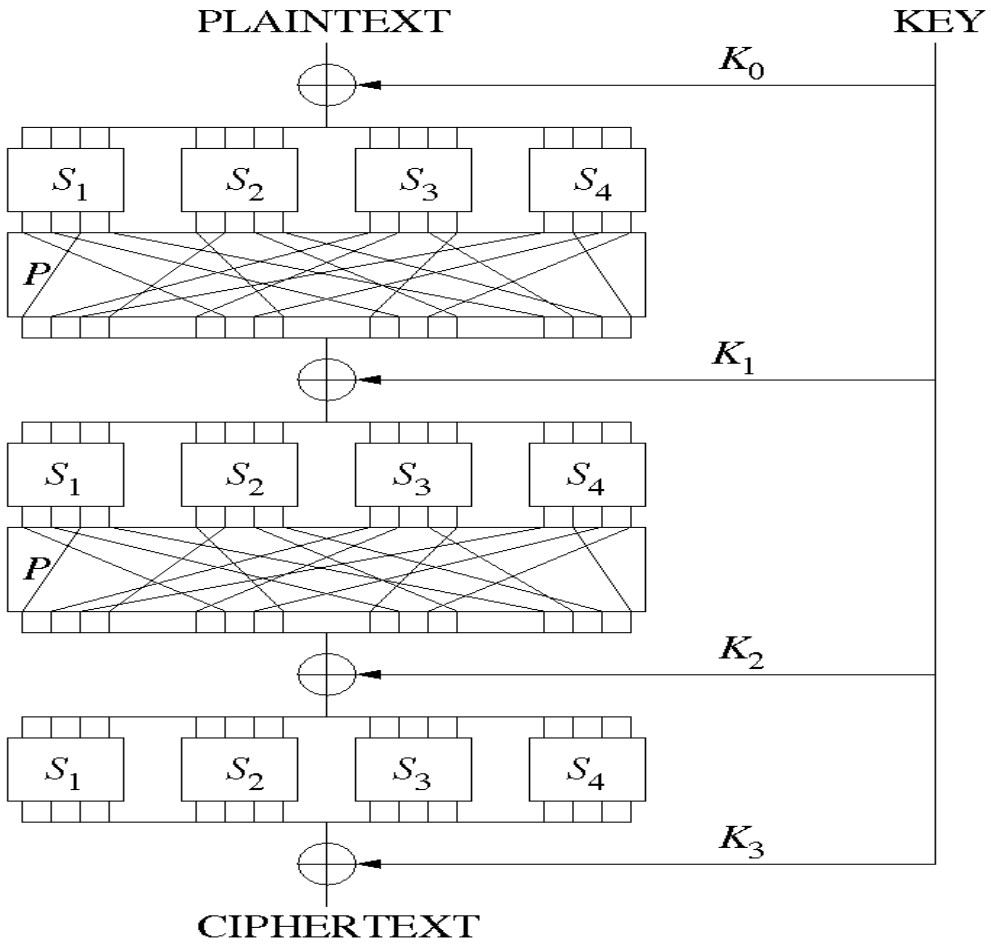
Libraries :

1.Tkinter

2.PDB

3.Function

Algorithm- Substitution–permutation network



In cryptography, an SP-network, or substitution–permutation network (SPN), is a series of linked mathematical operations used in block cipher algorithms such as AES (Rijndael), 3-Way, Kalyna, Kuznyechik, PRESENT, SAFER, SHARK, and Square.

Such a network takes a block of the plaintext and the key as inputs, and applies several alternating rounds or layers of substitution boxes (S-boxes) and permutation boxes (P-boxes) to produce the ciphertext block. The S-boxes and P-boxes transform (sub-)blocks of input bits into output bits. It is common for these transformations to be operations that are efficient to perform in hardware, such as exclusive or (XOR) and bitwise rotation. The key is introduced in each

round, usually in the form of "round keys" derived from it. (In some designs, the S-boxes themselves depend on the key.)

Decryption is done by simply reversing the process (using the inverses of the S-boxes and P-boxes and applying the round keys in reversed order).

Components:

An S-box substitutes a small block of bits (the input of the S-box) by another block of bits (the output of the S-box). This substitution should be one-to-one, to ensure invertibility (hence decryption). In particular, the length of the output should be the same as the length of the input (the picture on the right has S-boxes with 4 input and 4 output bits), which is different from S-boxes in general that could also change the length, as in Data Encryption Standard (DES), for example. An S-box is usually not simply a permutation of the bits. Rather, a good S-box will have the property that changing one input bit will change about half of the output bits (or an avalanche effect). It will also have the property that each output bit will depend on every input bit.

A P-box is a permutation of all the bits: it takes the outputs of all the S-boxes of one round, permutes the bits, and feeds them into the S-boxes of the next round. A good P-box has the property that the output bits of any S-box are distributed to as many S-box inputs as possible.

At each round, the round key (obtained from the key with some simple operations, for instance, using S-boxes and P-boxes) is combined using some group operation, typically XOR.

Properties :

A single typical S-box or a single P-box alone does not have much cryptographic strength: an S-box could be thought of as a substitution cipher, while a P-box could be thought of as a transposition cipher. However, a well-designed SP network with several alternating rounds of S- and P-boxes already satisfies Shannon's confusion and diffusion properties:

The reason for diffusion is the following: If one changes one bit of the plaintext, then it is fed into an S-box, whose output will change at several bits, then all these changes are distributed by the P-box among several S-boxes, hence the outputs of all of these S-boxes are again changed at several bits, and so on. Doing several rounds, each bit changes several times back and forth, therefore, by the end, the ciphertext has changed completely, in a pseudorandom manner. In particular, for a randomly chosen input block, if one flips the i-th bit, then the probability that the j-th output bit will change is approximately a half, for any i and j, which is the strict avalanche criterion. Vice versa, if one changes one bit of the ciphertext, then attempts to decrypt it, the result is a message completely different from the original plaintext—SP ciphers are not easily malleable.

The reason for confusion is exactly the same as for diffusion: changing one bit of the key changes several of the round keys, and every change in every round key diffuses over all the bits, changing the ciphertext in a very complex manner.

If an attacker somehow obtains one plaintext corresponding to one ciphertext—a known-plaintext attack, or worse, a chosen plaintext or chosen-ciphertext attack—the confusion and diffusion make it difficult for the attacker to recover the key.

Performance :

Although a Feistel network that uses S-boxes (such as DES) is quite similar to SP networks, there are some differences that make either this or that more applicable in certain situations. For a given amount of confusion and diffusion, an SP network has more "inherent parallelism"[1] and so — given a CPU with many execution units — can be computed faster than a Feistel network.[2] CPUs with few execution units — such as most smart cards — cannot take advantage of this inherent parallelism. Also SP ciphers require S-boxes to be invertible (to perform decryption); Feistel inner functions have no such restriction and can be constructed as one-way functions.

Tkinter

1.1 DEFINITION

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's *de facto* standard GUI. Tkinter is included with standard Linux, Microsoft Windows and macOS installs of Python.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

1.2 FEATURES

PyTorch provides two high-level features:

- **Tk(screenName=None, baseName=None, className='Tk', useTk=1):** To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

  - m=tkinter.Tk() where m is the name of the main window object

- **mainloop():** There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur, and process the event as long as the window is not closed.

  - m.mainloop()

PDB:

Debugging in Python is facilitated by pdb module (python debugger) which comes built-in to the Python standard library. It is actually defined as the class Pdb which internally makes use of bdb(basic debugger functions) and cmd (support for line-oriented command interpreters) modules. The major advantage of pdb is it runs purely in the command line, thereby making it great for debugging code on remote servers when we don't have the privilege of a GUI-based debugger.

pdb supports:

Setting breakpoints

Stepping through code

Source code listing

Viewing stack traces

13 REFERENCES

- https://www.geeksforgeeks.org/python-debugger-pythonpdb/
- https://www.nku.edu/~christensen/1402%20permutation%20ciphers.pdf
- https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking
- https://www.sailpoint.com/identity-library/five-types-of-cybersecurity/