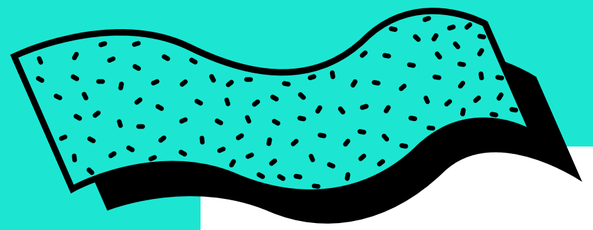
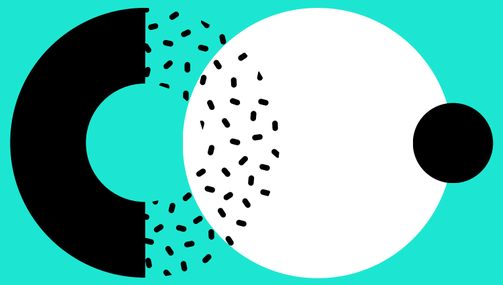


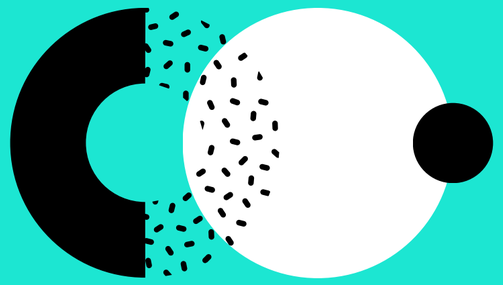
CODENAME: PROJECT BLOWFISH

presented by talal almajhad



SO WHAT IS BLOWFISH?

*Blowfish is a symmetric-key block cipher algorithm designed by Bruce Schneier in 1993.
* Blowfish uses a variable-length key between 32 and 448 bits.
*Blowfish is known for its simple design and fast encryption/decryption.
*Blowfish is considered a strong and secure encryption algorithm.

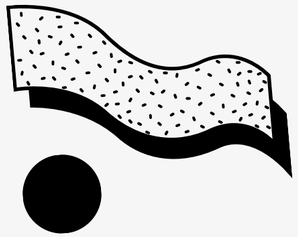


HOW DOES IT WORK

It works by first dividing the data into fixed-length blocks, usually 64 bits long. The key is then used to initialize a large substitution table (called a S-box) and a complex key expansion mechanism is used to generate a series of subkeys that are used to modify the S-box.

Next, each block is encrypted using the modified S-box and subkeys in multiple rounds. Each round involves substitution and permutation operations, which scramble the plaintext in a complicated way. The number of rounds depends on the key size and can range from 16 to 24.

The result is a highly secure and flexible encryption algorithm that can handle a wide range of data types and sizes.



PROS OF BLOWFISH

Widely supported

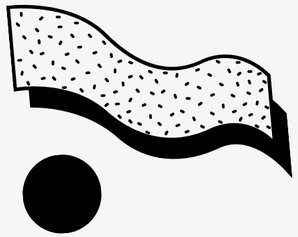
It is widely supported by many software applications, making it easy to integrate into different systems.

Strong encryption

It uses a variable-length key (ranging from 32-448 bits), making it possible to use very strong keys that are extremely difficult to crack.

Fast and efficient

Blowfish is designed to be fast and efficient, allowing for quick encryption and decryption of large amounts of data even on older systems.



CONS OF BLOWFISH

Vulnerable to brute-force attacks

Although the encryption key of Blowfish can be very strong, it may still be vulnerable to attacks from advanced brute-force cracking methods.

Key management

The key management for Blowfish can be a challenge since it can be difficult to change the encryption key once in use.

Old age

Blowfish was designed in 1993, which makes it relatively old compared to other modern encryption algorithms.

this is my blowfish cipher code

The code is using the `Blowfish` module from the `Crypto` library to encrypt and decrypt the message. The `Blowfish` cipher is a symmetric key block cipher. That means the same key is used to encrypt and decrypt the message.

```python

from Crypto.Cipher import Blowfish
import os
def encrypt_message(message, key):
    cipher = Blowfish.new(key, Blowfish.MODE_ECB)

    message = message.encode()
    message += b'\0' * (8 - len(message) % 8)

    ciphertext = cipher.encrypt(message)
    return ciphertext

def decrypt_message(ciphertext, key):
    decipher = Blowfish.new(key, Blowfish.MODE_ECB)
    message_decrypted = decipher.decrypt(ciphertext)
    message_decrypted = message_decrypted.rstrip(b'\0')
    message_decrypted = message_decrypted.decode()
    return message_decrypted

def input_hex(prompt):
    while True:
        try:
            value = input(prompt)
            value = bytes.fromhex(value)
            return value
        except ValueError:
            print("Invalid input format. Please enter a valid hexadecimal value.")

def encoding_decoding_message():
    encode_flag = input("Do you want to encrypt or decrypt a message? (Enter 'E' to encrypt, 'D' to decrypt): ").lower() == 'e'

    if encode_flag:
        message = input("Enter the message to encrypt: ")
        key = os.urandom(16)

        ciphertext = encrypt_message(message, key)

        print("Encrypted message:", ciphertext.hex())
        print("Key:", key.hex())

    else:
        key = input_hex("Enter the key: ")
        ciphertext = input_hex("Enter the ciphertext: ")

        message_decrypted = decrypt_message(ciphertext, key)

        print("Decrypted message:", message_decrypted)

encoding_decoding_message()

```

THE PROGRAM DEFINES TWO MAIN FUNCTIONS: `ENCRYPT_MESSAGE()` AND `DECRYPT_MESSAGE()`.

`encrypt_message()` takes a message and a key as input. It pad the message so it is a multiple of 8 bytes. Then it encrypts the message with the Blowfish cipher using the key. Then it returns the ciphertext.

`decrypt_message()` takes a ciphertext and a key as input. It decrypts the ciphertext with the Blowfish cipher using the key. Then it removes any padding from the decrypted message and returns the plaintext message.

TESTING THE CODE