

LAB ASSIGNMENT: 02

SUBMITTED BY:



SUBMITTED TO: MA’AM AMBREEN GUL

DATE: 12TH OCTOBER, 2025

COURSE: Information Security

***DES Block Cipher Algorithm***

*Graded Lab Task 1:* You have implemented DES there is built in implemented DES in python in crypto cipher module use it for encryption/decryption and provide output sample example

A screenshot of a computer program

AI-generated content may be incorrect.

**What This Code Does:**

* This code is a **secret message machine** that uses DES encryption. It takes a normal message, turns it into secret code using a password, and then turns it back to the original message.

**How It Works:**

**Step 1 - Make a Password:**

* Creates a random 8-character password (this is the DES key)

**Step 2 - Hide the Message (Encryption):**

* Takes your message and the password
* Adds some extra characters to make it the right size (called padding)
* Scrambles it into secret code that looks like random numbers

**Step 3 - Get Back Message (Decryption):**

* Takes the secret code and the same password
* Unscrambles the secret code
* Removes the extra characters
* Returns your original message

*Graded Lab Task 2:*

**1. Simple Step-by-Step Explanation**

**Step 1: The Attacker's Starting Point**

* **Attacker has:** A known message + its encrypted version
* **Example:**
  + Plaintext: "HELLO"
  + Ciphertext: "XJ9#L"

**Step 2: Split the Problem**

* DES uses 56-bit key (HUGE number: 72,057,594,037,927,936)
* Instead of trying all keys, split into two parts:
  + **K1** = First 28 bits
  + **K2** = Last 28 bits

**Step 3: Forward Attack (From Start)**

For each possible K1:

P → [Encrypt with K1] → I1

Store (I1, K1) in a table

**What happens:** Try all first-half keys and see where they lead

**Step 4: Backward Attack (From End)**

For each possible K2:

C → [Decrypt with K2] → I2

Check if I2 exists in table

**What happens:** Try all second-half keys and see where they come from

**Step 5: The "Meet" Moment**

When I1 == I2:

SUCCESS! Full Key = K1 + K2

**The magic:** Both paths meet at the same middle point!

**2. Real Example**

Let's say DES encryption works like this:

"HELLO" → [FIRST HALF] → "MID123" → [SECOND HALF] → "XJ9#L"

**Attack Process:**

ATTACKER TRIES:

K1 = "ABC" → "HELLO" → Encrypt → Gets "MID111"

K1 = "SEC" → "HELLO" → Encrypt → Gets "MID123"

K2 = "XYZ" → "XJ9#L" → Decrypt → Gets "MID999"

K2 = "RET" → "XJ9#L" → Decrypt → Gets "MID123"

MATCH FOUND! Both reached "MID123"

FULL KEY = "SEC" + "RET" = "SECRET"

**3. Why This is So Powerful**

**Normal Brute Force:**

Try all 56-bit keys one by one:

2⁵⁶ = 72,057,594,037,927,936 attempts

Time: YEARS!

**MITM Attack:**

Try all 28-bit K1 keys: 2²⁸ = 268,435,456 attempts

Try all 28-bit K2 keys: 2²⁸ = 268,435,456 attempts

Total: 536,870,912 attempts

Time: HOURS/DAYS!

**Speed Improvement:** About **134 million times faster!**

┌─────────────────────────────────────────────────────────────┐

│ MITM ATTACK FLOW │

│─────────────────────────────────────────────────────────────│

│ PLAINTEXT (P) CIPHERTEXT (C) │

│ ↓ ↓ │

│ ┌─────────┐ ┌─────────┐ │

│ │ GUESS │ │ GUESS │ │

│ │ ALL K1 │ │ ALL K2 │ │

│ │ KEYS │ │ KEYS │ │

│ └─────────┘ └─────────┘ │

│ ↓ ↓ │

│ ┌─────────┐ ┌─────────┐ │

│ │PARTIAL │ │PARTIAL │ │

│ │ENCRYPT │ │DECRYPT │ │

│ │WITH K1 │ │WITH K2 │ │

│ └─────────┘ └─────────┘ │

│ ↓ ↓ │

│ INTERMEDIATE I1 INTERMEDIATE I2 │

│ └─────────────┬──────────────┘ │

│ ↓ │

│ COMPARE I1 == I2? │

│ ↓ │

│ ┌───────────────┐ │

│ │ MATCH? │ │

│ └───────────────┘ │

│ ↓ │

│ ┌───────────────┐ │

│ │ FOUND KEY! │ │

│ │ K = K1 + K2 │ │

│ └───────────────┘ │

└─────────────────────────────────────────────────────────────┘