Domain – Telecom

focus – Optimization

Business challenge/requirement

LifeTel Telecom is the latest entrant in the highly competitive Telecom market of Singapore. It issues SIM to the verified users. Till now verification was manual through the photocopy of approved id card document. However, government has recently introduced Social ID called Reference ID which is mapped to fingerprint of user. LifeTel should now verify user against the fingerprint and Reference ID

Key issues

Build a system where when user enters Reference ID it is encrypted, so that hackers cannot view the mapping of Reference ID and finger print Considerations System should be secure

Considerations - System should be secure

Data volume - NA

Additional information - NA

Business benefits

Company will be able to quickly issue SIM to user and expected gain in volume is approximately 10 times as the manual process of verification is replaced with secure automated system

Approach to Solve

1. Read the input from command line – Reference ID

2. Check for validity – it should be 12 digits and allows on number and alphabet

3. Encrypt the Reference ID and print it for reference

import re

import base64

import os

# Function to validate Reference ID

def validate\_reference\_id(ref\_id):

if len(ref\_id) != 12 or not re.match("^[A-Za-z0-9]+$", ref\_id):

raise ValueError("Reference ID must be 12 alphanumeric characters.")

# Function to encrypt the Reference ID

def encrypt\_reference\_id(ref\_id, key):

# Create AES cipher

cipher = AES.new(key, AES.MODE\_EAX)

# Encrypt the Reference ID

ciphertext, tag = cipher.encrypt\_and\_digest(ref\_id.encode('utf-8'))

# Combine nonce, tag, and ciphertext for transmission/storage

return base64.b64encode(cipher.nonce + tag + ciphertext).decode('utf-8')

def main():

# Generate a secure random key (16 bytes for AES-128)

key = os.urandom(16)

# Read Reference ID from the user

ref\_id = input("Enter Reference ID (12 alphanumeric characters): ")

try:

# Validate Reference ID

validate\_reference\_id(ref\_id)