18) Write a python program for Bob uses the RSA cryptosystem with a very large modulus n for which the

factorization cannot be found in a reasonable amount of time. Suppose Alice sends a message to Bob by

representing each alphabetic character as an integer between 0 and 25 (A S 0, c, Z S 25) and then

encrypting each number separately using RSA with large e and large n. Is this method secure? If not,

describe the most efficient attack against this encryption method.

PROGRAM:-

```
e = 17
n = 3233  # In practice, this would be much larger

# Simulated message: "HELLO"
# A=0, B=1, ..., Z=25 => H=7, E=4, L=11, O=14
plaintext = [7, 4, 11, 11, 14]

# Encrypt each character separately
ciphertext = [pow(m, e, n) for m in plaintext]
print("Encrypted message:", ciphertext)

# Eve's attack: brute-force all possible plaintexts
lookup = {pow(m, e, n): m for m in range(26)}
print("\nLookup Table (cipher -> plain):", lookup)

# Eve decrypts the message
decrypted = [lookup[c] for c in ciphertext]
print("Decrypted message (as numbers):", decrypted)

# Optional: Convert numbers back to letters
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
decrypted_message = ''.join(alphabet[m] for m in decrypted)
print("Decrypted message (as letters):", decrypted_message)
```

OUTPUT:-

Encrypted message: [2369, 1387, 3061, 3061, 2549]

Lookup Table (cipher -> plain): {0: 0, 1: 1, 1752: 2, 1211: 3, 1387: 4, 3086: 5, 824: 6, 2369: 7, 2041: 8, 1972: 9, 1096: 10, 3061: 11, 1730: 12, 47: 13, 2549: 14, 3031: 15, 134: 16, 908: 17, 2100: 18, 615: 19, 3023: 20, 1188: 21, 2558: 22, 2037: 23, 1639: 24, 2211: 25}

Decrypted message (as numbers): [7, 4, 11, 11, 14]

Decrypted message (as letters): HELLO