Multi Time Pad is not Secure

BITS F463: Cryptography
AY 2022-23, II Semester

Dhruv Rawat (2019B3A70537P)

April 30, 2023

The Many-time Pad encryption is a type of symmetric-key encryption that relies on using the same one-time pad key for encrypting multiple plaintext messages. While this may seem like a straightforward way of encrypting data, it is inherently insecure and can be easily broken by attackers with sufficient resources.

The reason for this is that, in traditional one-time pad encryption, a random key is used only once to encrypt a single message, providing perfect secrecy. In many-time pad encryption, however, the same key is reused across multiple messages, which can lead to serious vulnerabilities in the encryption scheme. For example, if an attacker is able to obtain two or more ciphertexts that were encrypted using the same one-time pad key, they can easily obtain the XOR of the corresponding plaintexts, effectively breaking the encryption.

0.1 MTP Encryption:

Algorithm 1 MTP Encryption

```
with open(plain_text_file, "rb") as input_file, open(key_file, "rb") as key_file, open(cipher_text_file, "wb") as output_file:
while True: do
    input_byte = input_file.read(1)
    key_byte = key_file.read(1)
    if not input_byte or not key_byte: then
        break
    output_byte = bytes([ord(input_byte[0]) ^ ord(key_byte[0])])
    output_file.write(output_byte.hex().encode("utf-8"))
print(f"Ciphertext saved to file 'cipher_text_file' in hexadecimal format.")
```

0.2 MTP Decryption:

Algorithm 2 MTP Decryption

```
with open(cipher_text_file, "rb") as input_file, open(key_file, "rb") as key_file, open(plain_text_file, "wb") as output_file:
while True: do
    input_hex = input_file.read(2)
    if not input_hex then
        break
    input_byte = bytes.fromhex(input_file.decode("utf-8"))
    key_byte = key_file.read(1)
    if not input_byte or not key_byte: then
        break
    output_byte = bytes([input_byte[0] ^key_byte[0]])
    output_file.write(output_byte)
print(f'Plaintext saved to file 'plain_text_file'.")
```