OTP INVESTIGATION TASK

Laporan Ini dibuat untuk Memenuhi Salah Satu Tugas Evaluasi Tengah Semester

Pada Mata Kuliah Kriptografi

Dosen Pengampu: Kodrat Mahatma

Universitas Teknologi Digital

Disusun oleh:

Fernanda Syah Putra 20123019

PROGRAM STUDI S1 INFORMATIKA

UNIVERSITAS TEKNOLOGI DIGITAL

2025

Pengantar

Tugas ini memanfaatkan kerentanan *key reuse* pada One-Time Pad (OTP). Diberikan dua ciphertext yang dienkripsi dengan OTP menggunakan **kunci yang sama**, serta sebagian plaintext dari ciphertext pertama. Tujuan: menurunkan kunci OTP dari pasangan (C1, P1), kemudian menganalisis dampak kebocoran kunci terhadap ciphertext kedua (C2). Laporan ini berisi perhitungan langkah-demi-langkah, tabel konversi huruf→angka→kunci, hasil dekripsi C2, analisis konseptual, dan lampiran skrip Python untuk verifikasi di Google Colab.

Data:

- Ciphertext 1 (C1): `TLCYKUMGDFAWTZVOYKLENSZZHYZRW` (29 huruf)
- Plaintext sebagian (P1): hasil penghilangan spasi dan huruf kapital dari "MR JOHNSON LEFT HIS HOUSE LAST NIGHT" → `MRJOHNSONLEFTHISHOUSELASTNIGHT` → gunakan 29 huruf pertama: `MRJOHNSONLEFTHISHOUSELASTNIGH` (29 huruf)
- Ciphertext 2 (C2): `QXGLZMSOYTUVWSXKZVTLFQSKKMZXM` (29 huruf)

Konvensi: A→0, B→1, …, Z→25. Semua operasi di modulo 26.
Rumus utama: `K = (C1 - P1) mod 26`. Untuk dekripsi: `P2 = (C2 - K) mod 26`.

Perhitungan langkah-demi-langkah (tabel)

Tabel di bawah menunjukkan, untuk setiap posisi i (1..29): huruf C1, nilai numerik C1, huruf P1, nilai numerik P1, nilai kunci numerik, dan huruf kunci.

```
C1: TLCYKUMGDFAWTZVOYKLENSZZHYZRW
P1 (used): MRJOHNSONLEFTHISHOUSELASTNIGH
Derived key: HUTKDHUSQUWRASNWRWRMJHZHOLRLP
C2: QXGLZMSOYTUVWSXKZVTLFQSKKMZXM
Decrypted P2: JDNBWFYWIZYEWAKOIZCZWJTDWBIMX

Tabel posisi | C1 | C1# | P1 | P1# | K | K#
01 | T | 19 | M | 12 | H | 07
02 | L | 11 | R | 17 | U | 20
03 | C | 02 | J | 09 | T | 19
04 | Y | 24 | O | 14 | K | 10
05 | K | 10 | H | 07 | D | 03
06 | U | 20 | N | 13 | H | 07
07 | M | 12 | S | 18 | U | 20
08 | G | 06 | O | 14 | S | 18
09 | D | 03 | N | 13 | Q | 16
10 | F | 05 | L | 11 | U | 20
11 | A | 00 | E | 04 | W | 22
12 | W | 22 | F | 05 | R | 17
13 | T | 19 | T | 19 | A | 00
14 | Z | 25 | H | 07 | S | 18
15 | V | 21 | I | 08 | N | 13
16 | O | 14 | S | 18 | W | 22
17 | Y | 24 | H | 07 | R | 17
18 | K | 10 | O | 14 | W | 22
19 | L | 11 | U | 20 | R | 17
20 | E | 04 | S | 18 | M | 12
21 | N | 13 | E | 04 | J | 09
22 | S | 18 | L | 11 | H | 07
23 | Z | 25 | A | 00 | Z | 25
24 | Z | 25 | S | 18 | H | 07
25 | H | 07 | T | 19 | O | 14
26 | Y | 24 | N | 13 | L | 11
27 | Z | 25 | I | 08 | R | 17
28 | R | 17 | G | 06 | L | 11
29 | W | 22 | H | 07 | P | 15
```

Kunci (gabungan 29 huruf):

HUTKDHUSQUWRASNWRWRMJHZHOLRLP

Catatan: tabel dan kunci di atas dihitung dengan ketelitian digit per digit menggunakan konversi A=0…Z=25 dan operasi modulo 26.

Dekripsi Ciphertext 2 menggunakan kunci yang sama

Dengan kunci `HUTKDHUSQUWRASNWRWRMJHZHOLRLP`, dekripsi C2 menghasilkan plaintext berikut (29 huruf):

`JDNBWFYWIZYEWAKOIZCZWJTDWBIMX`

Hasil tersebut tidak membentuk kalimat bahasa Inggris yang bermakna. Kemungkinan penjelasan:

- C2 mungkin bukan hasil enkripsi plaintext bahasa Inggris alami, atau
- sumber soal memang memberi ciphertext acak untuk C2 inti analisis tetap sama: key reuse membuat C2 rentan jika P1 diketahui.

Ilustrasi algebraik konseptual (mengapa key reuse berbahaya):

Jika $C1 = P1 \oplus K$ dan $C2 = P2 \oplus K$, maka

$C1 \oplus C2 = (P1 \oplus K) \oplus (P2 \oplus K) = P1 \oplus P2$ (karena $K \oplus K = 0$).

Dengan mengetahui P1 sebagian, atau menebak kemungkinan kata/frase pada P1, penyerang dapat memperoleh fragmen P2. Contoh: jika penyerang tahu P1 penuh, K dapat dihitung langsung; lalu P2 = C2 ⊕ K.

Lampiran: Skrip Python

```
# ============================
# OTP Recovery Script - Verifikasi
# ============================

C1 = "TLCYKUMGDFAWTZVOYKLENSZZHYZRW"
P1 = "MRJOHNSONLEFTHISHOUSELASTNIGHT"[:29]
C2 = "QXGLZMSOYTUVWSXKZVTLFQSKKMZXM"


# --- Konversi string huruf A–Z menjadi angka 0–25 ---
def to_nums(s):
    return [ord(ch) - 65 for ch in s]

# --- Konversi angka 0–25 kembali ke huruf ---
def to_str(nums):
    return ''.join(chr((n % 26) + 65) for n in nums)


# --- Konversi C1 dan P1 ---
c1n = to_nums(C1)
p1n = to_nums(P1)

# --- Derive key K = C1 - P1 (mod 26) ---
kn = [(c - p) % 26 for c, p in zip(c1n, p1n)]
key = to_str(kn)

# --- Decrypt C2: P2 = C2 - K (mod 26) ---
c2n = to_nums(C2)
p2n = [(c - k) % 26 for c, k in zip(c2n, kn)]
```

```
p2 = to_str(p2n)

print("C1:", C1)
print("P1 (used):", P1)
print("Derived key:", key)
print("C2:", C2)
print("Decrypted P2:", p2)


# --- OPSIONAL: Tabel langkah-langkah ---
print("\nTabel posisi | C1 | C1# | P1 | P1# | K | K#")
for i, (c, p, k) in enumerate(zip(C1, P1, key),
start=1):
    print(f"{i:02d} | {c} | {ord(c)-65:02d} | {p} |
{ord(p)-65:02d} | {k} | {ord(k)-65:02d}")
```

Analisis konseptual

1. **Apa itu perfect secrecy?**

   Perfect secrecy (Shannon) berarti ciphertext tidak menyampaikan informasi apa pun tentang plaintext: pengetahuan tentang ciphertext tidak mengubah probabilitas distribusi plaintext. OTP dengan kunci acak sepadan panjang pesan, digunakan sekali, dan dirahasiakan sepenuhnya memenuhi definisi ini.

2. **Mengapa OTP dengan key reuse tidak aman?**

   Key reuse menghilangkan jaminan perfect secrecy karena dua ciphertext yang memakai kunci sama saling berkaitan: $C1 \oplus C2 = P1 \oplus P2$.

Dengan mengetahui sebagian P1 atau memanfaatkan statistik bahasa (frekuensi huruf, bigram, n-gram), penyerang dapat menebak P2 atau K.

3. **Apakah OTP aman jika kunci tidak acak?**

   Tidak. Keamanan OTP bergantung pada kunci yang benar-benar acak. Jika kunci dihasilkan dari PRNG lemah atau memiliki pola, penyerang dapat mengeksploitasi korelasi/kemungkinan untuk memulihkan kunci atau plaintext.

4. **Mengapa OTP jarang digunakan dalam kriptografi modern?**

   Meskipun aman secara teoretis, OTP tidak praktis: butuh kunci sepanjang pesan, masalah distribusi dan penyimpanan kunci, dan risiko operasional tinggi (reuse, kebocoran). Sebaliknya, cipher modern menggunakan kunci pendek yang dikelola melalui protokol yang aman.

5. **Bandingkan OTP dengan cipher modern (mis. AES)**

   - *Tingkat keamanan*: OTP (dengan kunci ideal) memberi perfect secrecy; AES memberi keamanan komputasional (dengan asumsi kekuatan matematis/algoritmik).
   - *Praktikalitas*: AES lebih efisien dan praktis (kunci 128/256 bit, enkripsi cepat).
   - *Manajemen kunci*: AES lebih mudah dioperasionalkan menggunakan protokol kunci publik/terpusat; OTP butuh kunci besar dan distribusi offline aman.