

* Tugas 3

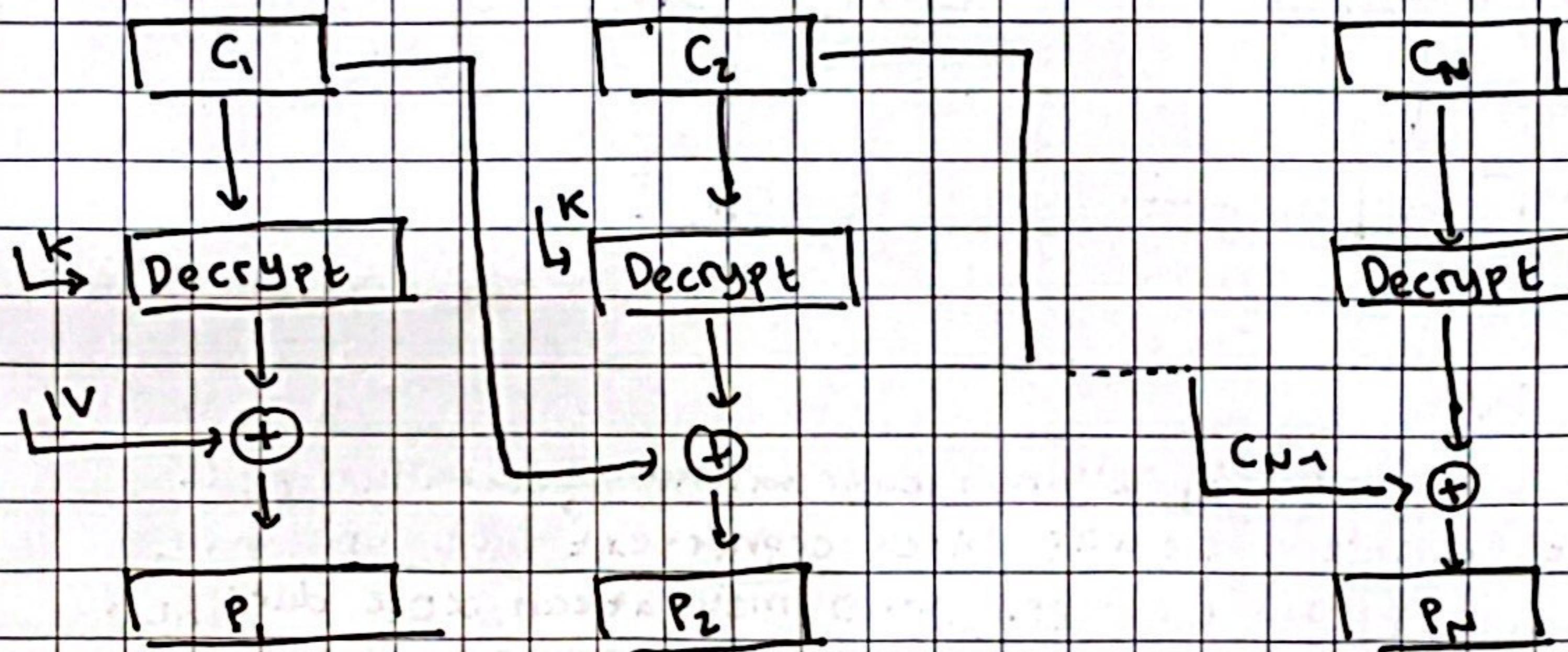
Anggota: 1. Adila Rahma Widja (221492834 / PA 721196)

2. Alexander Adam Mukhaer (221497621 / PA 121936)

3. Muhammad Hariish Hafiz (221504651 / PA 121712)

1.) Pada mode ECB, jika terjadi kesalahan pada sebuah blok ciphertext yang dikirimkan, hanya blok plaintext terkait yang terpengaruh. Namun, pada mode CBC, kesalahan ini merambat. Misalnya, kesalahan pada C_1 , yang dikirimkan jelas mensak P_1 dan P_2 (Lihat gambar ECB).

a. Apakah ada blok selain P_2 yang terkena dampak?



$$P_1 = D(K, C_1) \oplus IV$$

↳ Block plaintext pertama akan terkena dampak langsung karena setiap block plaintext dihasilkan dengan mendekripsi block ciphertext (C_1) dan meng-XOR hasilnya dengan IV. Dengan begitu jika C_1 salah maka P_1 juga akan salah

$$P_2 = D(K, C_2) \oplus C_1$$

↳ Block plaintext kedua akan terkena dampak juga oleh C_1 karena dalam proses dekripsi untuk P_2 terdapat hubungan dengan C_1 . Maka P_2 juga akan salah, meskipun C_2 benar

$$P_3 = D(K, C_3) \oplus C_2$$

$$P_4 = D(K, C_4) \oplus C_3$$

$$P_5 = D(K, C_5) \oplus C_4$$

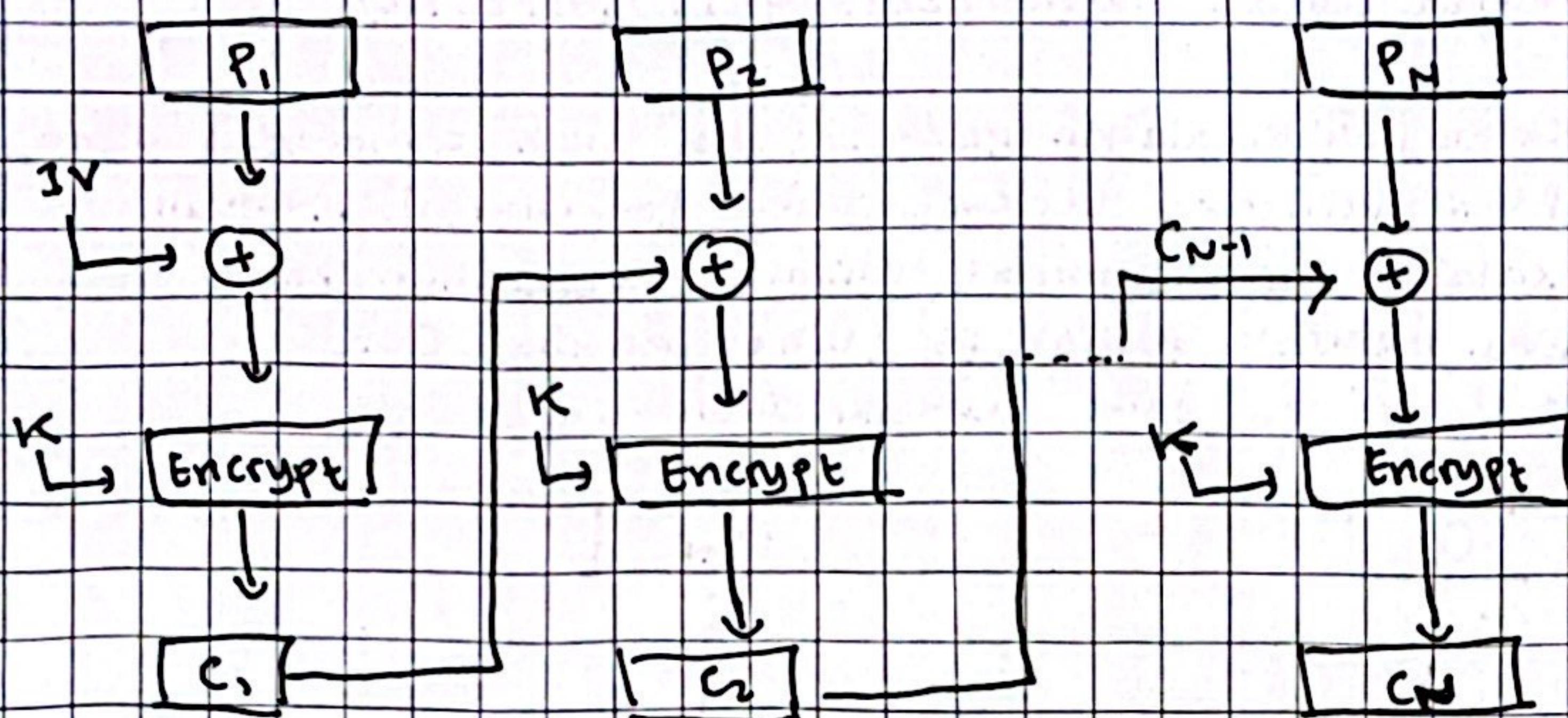
:

$$P_n = D(K, C_n) \oplus C_{n-1}$$

Pada block plaintext setelah P_2 tidak akan terkena dampak dari kesalahan C_1 . Karena pada block plaintext setelah P_2 tidak ada hubungannya dengan C_1 , sehingga kesalahan C_1 tidak dapat mensak P_3, P_4, P_5, \dots

Dengan begitu, dapat disimpulkan terdapat block selain P_2 yang terkena dampak, yaitu P_1 (plaintext pertama)

b. Misalnya ada error satu bit pada versi sumber plaintext P_1 . Ke berapa banyak block ciphertext error ini merambat? Apa dampaknya bagi penerima (receiver)?



$$\Rightarrow C_1 = E(K, P_1 \oplus IV)$$

↳ kesalahan satu bit pada P_1 akan menyebabkan perubahan pada blok ciphertext C_1 . Karena ciphertext pertama (C_1) pada CBC yang dihasilkan melalui proses enkripsi, yaitu plaintext pertama di XOR dengan IV dan kemudian dienkripsi dengan K sehingga menghasilkan C_1 . Dengan begitu kesalahan P_1 membuat C_1 juga salah.

$$C_2 = E(K, P_2 \oplus C_1)$$

↳ karena C_1 salah (yang drakibatkan kesalahan P_1). Maka dalam menentukan C_2 juga akan salah yang dikarenakan proses enkripsi kedua melibatkan C_1 dalam prosesnya

$$C_3 = E(K, P_3 \oplus C_2)$$

↳ karena C_2 salah. Maka dalam menentukan C_3 juga akan salah yang dikarenakan proses enkripsi kedua melibatkan C_2 dalam prosesnya. Nantinya kesalahan ini akan terus merambat hingga menentukan ciphertext terakhir. Dengan begitu, banyaknya block ciphertext error ini akan merambat pada seluruh proses enkripsi.

⇒ Dampak bagi penerima (receiver)

↳ ketika receiver mencoba mendeskripsikan blok-blok ciphertext yang diterima maka:

- ① Dekripsi $C_1 \Rightarrow P_1 = D(K, C_1) \oplus IV$. Karena dalam prosesnya melibatkan C_1 dan C_1 juga salah maka P_1 akan salah juga
- ② Dekripsi $C_2 \Rightarrow P_2 = D(K, C_2) \oplus C_1$. Karena dalam prosesnya melibatkan C_1 & C_1 yang juga salah keduanya maka P_2 akan salah juga.
Begini juga pada ciphertext selanjutnya karena masing-masing blok sebelumnya salah. Dengan begitu, semua blok plaintext yang dideskripsi selanjutnya juga akan salah.

Kesimpulan: Receiver tidak akan menerima satupun dekripsi yang benar. Maka receiver tidak akan menerima plaintext yang sama/tepat yang dikirim oleh sender

2.) Apakah mungkin melakukan operasi enkripsi secara paralel pada beberapa blok plaintext pada mode CBC? Bagaimana dengan deskripsi?

o) Enkripsi

$$C_j = E(K, [P_j \oplus C_{j-1}]), j=2, \dots, n$$

Setiap blok plaintext P_j harus di-XOR dengan blok ciphertext sebelumnya C_{j-1} sebelum di-enkripsi. Karena setiap blok ciphertext bergantung pada blok ciphertext sebelumnya. Maka proses enkripsi tidak bisa dilakukan secara paralel. Setiap blok harus di-enkripsi secara berurutan setelah blok sebelumnya selesai di-enkripsi.

o) Dekripsi

$$P_j = D(K, C_j) \oplus C_{j-1}, j=2, \dots, n$$

Setiap blok ciphertext C_j dapat di-dekripsi secara paralel. Meskipun apabila ingin menghasilkan plaintext, harus meng-XOR kan dengan ciphertext sebelumnya, akan tetapi dalam operasi dekripsi setiap blok hanya membutuhkan masing-masing ciphertext dan K untuk mendekripsi kanya.

3. [40 points] Alice dan Bob setuju berkomunikasi secara pribadi melalui email menggunakan skema berdasarkan RC4, namun mereka ingin menghindari penggunaan kunci rahasia baru untuk setiap transmisi. Alice dan Bob secara pribadi menyetujui kunci 128-bit k . Untuk mengenkripsi pesan m , yang terdiri sebuah bit string, prosedur berikut digunakan.

1. Pilih sebuah nilai acak 64-bit v .
2. Hasilkan ciphertext $c = \text{RC4}(v \parallel k) \oplus m$.
3. Kirimkan string bit $(v \parallel c)$.

Catatan: $(a \parallel b)$ bermakna string bit a disambungkan dengan string bit b .

- a. Misalkan Alice menggunakan prosedur ini untuk mengirim pesan m ke Bob. Jelaskan bagaimana Bob dapat memperoleh kembali pesan m dari $(v \parallel c)$ menggunakan k .
- b. Jika musuh mengamati beberapa pasangan nilai $(v_1 \parallel c_1), (v_2 \parallel c_2), \dots$ yang dikirimkan antara Alice dan Bob, bagaimana dia dapat menentukan kapan *key stream* yang sama telah digunakan untuk mengenkripsi dua pesan?
- c. Kira-kira berapa banyak pesan yang dapat Alice kirimkan sebelum *key stream* yang sama digunakan dua kali?
- d. Apa implikasinya terhadap masa pakai kunci k (yaitu jumlah pesan yang dapat dienkripsi menggunakan k)?

③ a.) Bob mengembalikan pesan m dari $(v \parallel c)$ menggunakan k

Proses enkripsi oleh Alice

1. Alice memilih nilai acak 64-bit v
2. Alice menyambungkan v dengan secret key k sehingga membentuk $(v \parallel k)$
3. Alice menggunakan **RC4** untuk menghasilkan *key stream* = $\text{RC4}(v \parallel k)$
4. Alice melakukan operasi **XOR** antara pesan m dan *key stream* untuk menghasilkan c

$$c = \text{key stream} \oplus m$$

5. Alice mengirimkan string bit gabungan $(v \parallel c)$ ke Bob

Proses dekripsi oleh Bob.

1. Bob menerima string bit gabungan $(v \parallel c)$ dari Alice.
2. Bob memisahkan v dan c :
 - v = 64-bit pertama dari $(v \parallel c)$
 - c = sisa bit dari $(v \parallel c)$
3. Bob menyambungkan v dengan private key k sehingga membentuk $(v \parallel k)$
4. Bob menggunakan **RC4** untuk menghasilkan *key stream* yang sama dengan Alice

$$\text{Key Stream} = \text{RC4}(v \parallel k)$$

5. Bob menghasilkan pesan asli m dengan melakukan operasi **XOR** antara ciphertext dan *key stream*

b.) Menentukan kepan key stream yang sama digunakan untuk mengenkripsi dua pesan.

Jika musuh mengetahui beberapa pasang nilai $(V_1 \parallel C_1), (V_2 \parallel C_2), \dots$:

1. Musuh mencari nilai acak v dari setiap pesan $(V \parallel c)$
2. Musuh mencari apakah terdapat dua pesan $(V_i \parallel C_i) (V_j \parallel C_j)$ di mana $V_i = v_j$. Jika berdapat dua pesan dengan $V_i = V_j$, maka key stream yang dihasilkan dari $RC4(V_i \parallel K)$ akan sama untuk kedua pesan tersebut karena input ke fungsi $RC4$ nya sama.
3. Dengan key stream yang sama, musuh dapat melakukan **XOR** pada ciphertext C_i dan C_j :

$$C_i \oplus C_j = (\text{key stream} \oplus m_i) \oplus (\text{key stream} \oplus m_j) = m_i \oplus m_j$$

Operasi ini menghilangkan key stream dan menghasilkan **XOR** dari dua pesan asli m_i & m_j . Jika musuh mengetahui salah satu pesan, ia dengan mudah mendapat pesan lainnya.

c.) Berapa banyak pesan yang dapat Alice kirimkan sebelum key stream yang sama digunakan dua kali? Vektor v adalah nilai acak 64-bit. Sehingga ada 2^{64} kemungkinan untuk nilai v . Probabilitas bentrok (collision) dapat diperkirakan menggunakan aproksimasi hari ulang tahun (Birthday paradox).

Menurut Birthday Paradox, kemungkinan bentrok signifikan setelah mengamati sekitar $\sqrt{2^{64}}$ pesan. Ini berarti setelah sekitar 9 miliar pesan, kemungkinan ada dua nilai v yang sama menjadi signifikan.

d.) Implikasi terhadap masa pakai kunci K (jumlah pesan yang dapat dienkripsi menggunakan kunci K)

Mengingat bahwa probabilitas bentrok menjadi signifikan setelah sekitar 2^{32} pesan, kunci K sebaiknya diganti sebelum mencapai jumlah pesan tersebut untuk menghindari resiko kerumahan.

Jadi, jumlah maksimum pesan yang dapat dienkripsi menggunakan satu kunci K adalah sekitar 2^{32} . Dengan demikian, untuk menjaga keamanan komunikasi, Alice dan Bob harus mengganti kunci K sebelum mengirim sekitar 2^{32} pesan untuk mencegah penggunaan ulang key stream yang dapat dikripsi oleh musuh.

4) a) Berapa periode maksimum yang diperoleh dari generator berikut?

$$X_{n+1} = (aX_n) \bmod 2^4$$

Jawab: Untuk generator linear kongruen dengan modulus $m = 2^4 = 16$ dan $c = 0$, periode maksimum yang dapat diperoleh adalah $P = \frac{m}{q} = \frac{16}{4} = 4$ //

b) Berapa seharusnya nilai a ?

Jawab: Untuk mencapai periode maksimum $P = 4$, nilai a harus dalam bentuk $3+8k$ atau $5+8k$ untuk $k = 0, 1, \dots$

→ Jika $k = 0$:

$$\begin{aligned} \circ \quad a &= 3+8 \cdot 0 = 3 \\ \circ \quad a &= 5+8 \cdot 0 = 5 \end{aligned}$$

→ Jika $k = 1$:

$$\begin{aligned} \circ \quad a &= 3+8 \cdot 1 = 11 \\ \circ \quad a &= 5+8 \cdot 1 = 13 \end{aligned}$$

∴ Jadi, nilai a yang memenuhi syarat adalah 3, 5, 11, atau 13 //

c) Pembatasan apa yang diperlukan pada seed?

Jawab: Untuk mencapai periode maksimum $P = 4$, seed X_0 harus ganjil. Oleh karena itu, pembatasan pada seed adalah seed harus berupa bilangan ganjil dalam rentang 1 sampai 15.

∴ Sehingga, seed X_0 harus termasuk dalam himpunan:

$$X_0 \in \{1, 3, 5, 7, 9, 11, 13, 15\}$$

5.) a. Hitung koefisien Bézout dari 978 dan 1056 dengan menggunakan algoritma Euclid. Perlihatkan perhitungan anda.

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

$$\therefore \gcd(978, 1056)$$

$$i. 1056 = (1) \cdot 978 + 78$$

$$ii. 978 = (12) \cdot 78 + 42$$

$$iii. 78 = (1) \cdot 42 + 36$$

$$\Rightarrow \gcd(978, 1056) = b$$

$$iv. 42 = (1) \cdot 36 + 6$$

$$v. 36 = (6) \cdot 6 + 0$$

→ Dari iv dan iii di atas:

$$6 = 42 - (1) \cdot 36 \quad \dots iv$$

$$36 = 78 - (1) \cdot 42 \quad \dots iii$$

* Dengan mensubstitusi persamaan ke-3 ke persamaan ke-4:

$$6 = 42 - (1)(78 - (1) \cdot 42)$$

$$6 = (2)42 - (1)78 \dots vi$$

→ Dari vi dan ii

$$6 = (2)42 - (1)78 \dots vi$$

$$42 = 978 - (12)78 \dots ii$$

* Dengan mensubstitusi persamaan ke-2 ke persamaan ke-6:

$$6 = (2)(978 - (12)78) - (1)78$$

$$6 = (2)978 - (25)78 \dots vii$$

→ Dari vii dan i

$$6 = (2)978 - (25)78 \dots vii$$

$$78 = 1056 - (1)978 \dots i$$

* Dengan mensubstitusi persamaan ke-1 ke persamaan ke-7:

$$6 = (2)978 - 25(1056 - (1)978)$$

$$6 = (27)978 - (25)1056$$

$$\therefore \gcd(978, 1056) = s \cdot a + t \cdot b$$

$$b = (27)978 - (25)1056$$

Jadi, koefisien Bézout dari 978 dan 1056 adalah $s = 27$ dan $t = -25$

b. Tentukan invers dari 977 pada modulo 1056 dengan menggunakan algoritma Euclid. Perlihatkan perhitungan anda!

$$1056 = (1) \cdot 977 + 79$$

Menelusuri dari belakang:

$$977 = (12) \cdot 79 + 29$$

$$1 = 3 - (1) \cdot 2$$

$$79 = (2) \cdot 29 + 21$$

$$1 = 3 - (1)(5 - (1) \cdot 3) \Rightarrow 1 = (2) \cdot 3 - 5$$

$$29 = (1) \cdot 21 + 8$$

$$1 = (2)(8 - 5) - 5 \Rightarrow 1 = (2) \cdot 8 - (3) \cdot 5$$

$$21 = (2) \cdot 8 + 5$$

$$1 = (2) \cdot 8 - (3)(21 - (2) \cdot 8) \Rightarrow 1 = (8)8 - (3) \cdot 21$$

$$8 = (1) \cdot 5 + 3$$

$$1 = (8)(29 - (1) \cdot 21) - (3) \cdot 21 \Rightarrow (8) \cdot 29 - (11) \cdot 21$$

$$5 = (1) \cdot 3 + 2$$

$$1 = (8) \cdot 29 - 11(79 - (3) \cdot 29) \Rightarrow 1 = (30) \cdot 29 - (11) \cdot 79$$

$$3 = (1) \cdot 2 + 1$$

$$1 = (30) \cdot 977 - (12) \cdot 79 - (11) \cdot 79 \Rightarrow 1 = (30) \cdot 977 - (37) \cdot 79$$

$$2 = (1) \cdot 1 + 0$$

$$1 = (30) \cdot 977 - (37) \cdot (1056 - (1) \cdot 977) = (401) \cdot 977 - (37) \cdot 1056$$

$\text{GCD} = 1$, maka ada invers

karena $977 \cdot 401 \equiv 1 \pmod{1056}$ maka invers dari

dari 977 modulo 1056 adalah 401,

c. Tentukan semua akar primitif dari modulo 17. Perlihatkan dalam bentuk tabel

| a | a^2 | a^3 | a^4 | a^5 | a^6 | a^7 | a^8 | a^9 | a^{10} | a^{11} | a^{12} | a^{13} | a^{14} | a^{15} | a^{16} |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|----------|----------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 9 | 8 | 16 | 15 | 15 | 9 | 1 | 2 | 4 | 8 | 16 | 15 | 13 | 9 | 1 |
| \checkmark 3 | 9 | 10 | 13 | 5 | 15 | 11 | 16 | 14 | 8 | 7 | 9 | 12 | 2 | 6 | 1 |
| 4 | 16 | 13 | 1 | 9 | 16 | 13 | 1 | 9 | 16 | 13 | 1 | 9 | 16 | 13 | 1 |
| \checkmark 5 | 8 | 6 | 13 | 19 | 2 | 10 | 16 | 12 | 9 | 11 | 9 | 3 | 15 | 7 | 1 |
| \checkmark 6 | 2 | 12 | 4 | 7 | 8 | 19 | 16 | 11 | 15 | 5 | 13 | 10 | 9 | 3 | 1 |
| \checkmark 7 | 15 | 3 | 4 | 11 | 9 | 12 | 16 | 10 | 2 | 19 | 13 | 6 | 8 | 5 | 1 |
| 8 | 13 | 2 | 16 | 9 | 4 | 15 | 1 | 8 | 13 | 2 | 16 | 9 | 4 | 15 | 1 |
| 9 | 13 | 15 | 16 | 8 | 4 | 2 | 1 | 9 | 13 | 15 | 16 | 8 | 9 | 2 | 1 |
| \checkmark 10 | 15 | 19 | 4 | 6 | 9 | 5 | 16 | 7 | 2 | 3 | 13 | 11 | 8 | 12 | 1 |
| \checkmark 11 | 2 | 5 | 9 | 10 | 8 | 3 | 16 | 6 | 15 | 12 | 13 | 7 | 9 | 19 | 1 |
| \checkmark 12 | 8 | 11 | 13 | 3 | 2 | 7 | 16 | 5 | 9 | 6 | 9 | 19 | 15 | 10 | 1 |
| 13 | 16 | 4 | 1 | 13 | 16 | 9 | 1 | 13 | 16 | 4 | 1 | 13 | 6 | 9 | 1 |
| \checkmark 14 | 9 | 7 | 15 | 12 | 15 | 6 | 16 | 3 | 8 | 10 | 4 | 5 | 2 | 11 | 1 |
| 15 | 9 | 9 | 16 | 2 | 13 | 8 | 1 | 15 | 4 | 9 | 16 | 2 | 13 | 8 | 1 |
| 16 | 1 | 15 | 1 | 16 | 1 | 16 | 1 | 16 | 1 | 16 | 1 | 16 | 1 | 16 | 1 |

Berdasarkan tabel di atas, dapat dilihat bahwa $3, 5, 6, 7, 10, 11, 12, 14$ merupakan akar-akar primitif dari modulo 17, karena untuk pangkat dari pangkat $1-16$ yang di modulo 17 menghasilkan semua angka dari $1-16$ tanpa adanya redundansi.