Problem 1

110550143 洪巧芸

- o (a) Compress then encrypt.
- o (b) Encrypt then compress.
- o (c) The order does not matter – either one is fine.
- x (d) The order does not matter – neither one will compress the data.

(a) 可減少加密時間，成本亦可減少頻寬
e.g. SSL - 資料被分成多個 fragments → 壓縮 fragments → 用 secret key 加密 fragment / Winrar
(b) 可避免壓縮攻擊，保留加密的 metadata
(c)(d) >< (a),(b)

Problem 2

- x (a) $G'(k) = G(k) \| G(k)$
- o (b) $G'(k) = G(k \oplus 1^s)$
- x (c) $G'(k) = G(0)$
- x (d) $G'(k) = G(1)$
- x (e) $G'(k) = G(k) \| 0$
- o (f) $G'(k_1, k_2) = G(k_1) \| G(k_2)$
- o (g) $G'(k) = \text{reverse}(G(k))$
- o (h) $G'(k) = rotation_n(G(k))$

(a) ∵ 只是串聯兩個自己 i.e 非 random ×
(b) 和 OPT 的概念相同
(c)(d) ∵ G(0), G(1) 是固定的，非 random ×
(e) >< (a)，只是變成串聯自己和 0 i.e 非 random ×
(f) 獨立的 secure PRG 串聯後它的 0,1 分佈還會是 random 的
(g)(h) 是 random 的 (∵ reverse, rotate 都不會影響 0,1 分佈

Problem 3

- × (a) $p_1 = (k_1, k_2), p_2 = (k_1, k_2), p_3 = (k_2')$
- × (b) $p_1 = (k_1, k_2), p_2 = (k_1', k_2'), p_3 = (k_2')$
- o (c) $p_1 = (k_1, k_2), p_2 = (k_1', k_2), p_3 = (k_2')$
- × (d) $p_1 = (k_1, k_2), p_2 = (k_2, k_2'), p_3 = (k_2')$
- × (e) $p_1 = (k_1, k_2), p_2 = (k_1'), p_3 = (k_2')$

要 $k_1 \oplus k_1'$ 或 $k_2 \oplus k_2'$

(a) $p_1, p_2 \to$ ×

(b) $p_2, p_3 \to$ ×

(c)
- $p_1, p_2 \to k_1 \oplus k_1'$
- $p_1, p_3 \to k_2 \oplus k_2'$ ✓
- $p_2, p_3 \to k_2 \oplus k_2'$

(d) p_2 can single decrypt ×

(e) $p_2, p_3 \to$ ×

Problem 4

- o (a) No, there is a simple attack on this cipher.
- × (b) Yes
- o (c) No, only the One Time Pad has perfect secrecy.

$E(k,m) = m + k \bmod 256$ $\{0,1,\cdots,256\}$

$D(k,c) = c - k \bmod 256$

(a) 可以直接用窮局分析找明文

(b) 承 (a)

(c) 目前是只有 One Time Pad 可以

Problem 5

- × (a) E'(k, m) = E(0^n, m)
- o (b) E'((k, k'), m) = E(k, m) ‖ E(k', m)
- × (c) E'(k, m) = E(k, m) ‖ MSB(m)
- o (d) E'(k, m) = 0 ‖ E(k, m) (i.e. prepend 0 to the ciphertext)
- × (e) E'(k, m) = E(k, m) ‖ k
- o (f) E'(k, m) = reverse(E(k, m))
- o (g) E'(k, m) = $rotation_n$(E(k, m))

破解 → 要求加密 0^n 和 1^n

× (a) 已知 key 為 0^n

(b) 拆成兩部份不會洩漏 plaintext 的資訊

× (c) 要求加密 0^n 和 0^n-1 則可區分 EXP(0), EXP(1)

(d) 串聯 0 不會改變 E i.e. 不會洩漏 plaintext 的資訊

× (e) ∵是直接串接 key i.e. key 可直接從密文讀取

(f) (g) reverse 和 rotate 都不會洩漏 plaintext 的資訊

Problem 6

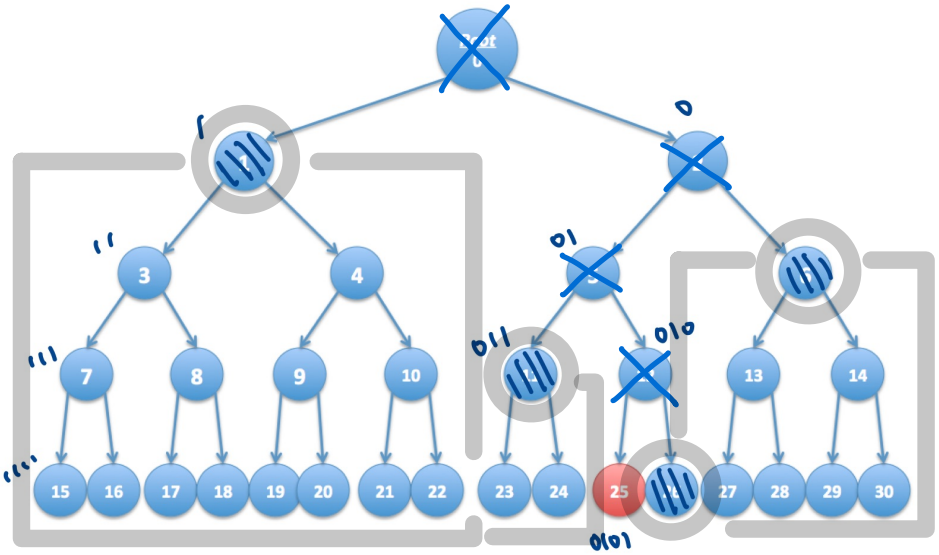
attack at dawn → 6c73d5240a948c86981bc294814d

a	61	0110 0001	0110 1100	6c	0000	1101	0	d
t	74	0111 0100	0111 0011	73	0000	0111	0	7
t	74	0111 0100	1101 0101	d5	1010	0001	a	1
a	61	0110 0001	0010 0100	24	0100	0101	4	5
c	63	0110 0011	0000 1010	0a	0110	1001	6	9
k	6B	0110 1011	1001 0100	94	1111	1111	f	f
	20	0010 0000	0100 1100	8c	0101	1100	a	c
a	61	0110 0001	1000 0110	86	1110	0111	e	7
t	74	0111 0100	1001 1000	98	1110	1100	e	c
	20	0010 0000	0001 1011	1b	0011	1011	3	b
d	64	0110 0100	1100 0010	c2	1010	0110	a	6
a	61	0110 0001	1001 0100	94	1111	0101	f	5
w	77	0111 0111	1000 0001	81	1111	0110	f	6
n	6E	0110 1110	0100 1101	4d	0010	0011	>3	
							L	

d	64	0110 0100	0000 1101	0110	1001	69	
e	65	0110 0101	0000 0111	0110	0010	62	
f	66	0110 0110	1010 0001	1100	0111	c7	
e	65	0110 0101	0100 0101	0010	0010	>0	
n	6e	0110 1110	0110 1001	0000	0111	07	
d	64	0110 0100	1111 1111	1001	1011	9b	6962c72007a9b8c8698bc89a994d#

a	和題目一樣	8c					
t		86					
		98					
		1b					
n	6e	0110 1110	1010 0110	1100	1000	c8	
o	6f	0110 1111	1111 0101	1001	1010	9a	
o	6f	0110 1111	1111 0110	1001	1001	99	
n	6e	0110 1110	0010 0011	0100	1101	4d	

Problem 1)



要包含 25 以外所有 leaf node → 不能包含 25 的 parent node ×
→ 每個 key 包含的子節點越多越好 (i.e. 越接近 root)
→ key 不要互相是 parent 和 child (∵ child 就不需要了)

⇒ 1, 6, 11, 26 #

Extra Credit

SHA-256 and SHA-512-truncated-to-256-bits
如果只要取 256 bits 的話我認為 SHA-256 較好,
因為它在編碼時所考慮的就是輸出 256 bits,
且截斷輸出的 hash 可能會影響它的屬性 (eg. collision resistance, pre-image resistance)